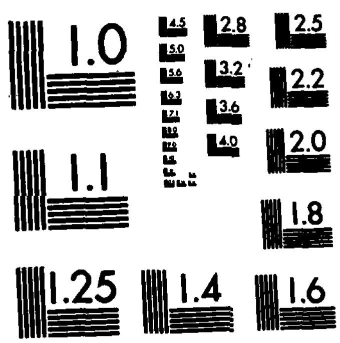
AD-A124 643 ALGORITHMS FOR DIFFERENTIAL GAMES WITH BOUNDED CONTROL AND STATES(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE A CHOMPAISAL MAR 82 1/2

UNCLASSIFIED DASG60-80-C-0007 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(2)

REPORT NUMBER: None	2. GOVT ACCESSION NO.: AD-A124643	3. RECIPIENT'S CATALOG NUMBER
TITLE (and Subtitle): ALGORITHMS FOR DIFFERENTIAL GAMES WITH BOUNDED CONTROL AND STATES		5. TYPE OF REPORT & PERIOD COVERED: Final, 11/29/79-11/28/81
		6. PERFORMING ORG. REPORT NUMBER: None
AUTHOR(s): Aran Chompaisal, UCLA, School of Engineering and Applied Science		8. CONTRACT OR GRANT NUMBER(s): DASG-60-80-C-0007
PERFORMING ORGANIZATION NAME AND ADDRESS: UCLA, School of Engineering and Applied Science, Los Angeles, California 90024		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: None
CONTROLLING OFFICE NAME AND ADDRESS: Advanced Technology Center, U.S. Army Ballistic Missile Defense Command, Huntsville, Alabama		12. REPORT DATE: March, 1982
		13. NUMBER OF PAGES: 151
MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same as No. 11		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

~~Distribution of this document is not limited.~~

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

None

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See attached

ABSTRACT

Pursuit and Evasion problems are probably the most natural application of differential game theory and have been treated by many authors as such. Very few problems of this class can be solved analytically. Fast and efficient numerical algorithm is needed to solve for an optimal or near optimal solution of a realistic pursuit and evasion differential game.

Some headways have been made in the development of numerical algorithm for this purpose. Most researchers, however, worked under an assumption that a saddle point exists for their differential game. Here, it is shown via two examples and a nonlinear stochastic differential game that such is not the case.

A first-order algorithm for computing an optimal control for each player, subject to control and/or state constraints, is developed without the assumption of saddle point existence. It is shown that a linear quadratic differential game with control and/or state constraints generally cannot be solved analytically. One such problem is developed and solved by the above algorithm. A new rationalization is offered in formulating a missile anti-missile problem as a nonlinear stochastic differential game. The algorithm developed here together with a convergence control method introduced by Jarmark is used to solve the missile anti-missile problem with fast computation time.

ALGORITHMS FOR DIFFERENTIAL GAMES
WITH BOUNDED CONTROL AND STATES

By

Aran Chompaisal

March, 1982

Submitted under contract
DASG-60-80-C-0007
Advances In Technology Development
for Exoatmospheric Intercept Systems

U.S. Army Ballistic Missile
Defense Command
Advanced Technology Center
Huntsville, Alabama

School of Engineering and Applied Science
University of California
Los Angeles, California

ABSTRACT

Pursuit and Evasion problems are probably the most natural application of differential game theory and have been treated by many authors as such. Very few problems of this class can be solved analytically. Fast and efficient numerical algorithm is needed to solve for an optimal or near optimal solution of a realistic pursuit and evasion differential game.

Some headways have been made in the development of numerical algorithm for this purpose. Most researchers, however, worked under an assumption that a saddle point exists for their differential game. Here, it is shown via two examples and a nonlinear stochastic differential game that such is not the case.

A first-order algorithm for computing an optimal control for each player, subject to control and/or state constraints, is developed without the assumption of saddle point existence. It is shown that a linear quadratic differential game with control and/or state constraints generally cannot be solved analytically. One such problem is developed and solved by the above algorithm. A new rationalization is offered in formulating a missile anti-missile problem as a nonlinear stochastic differential game. The algorithm developed here together with a convergence control method introduced by Jarmark is used to solve the missile anti-missile problem with fast computation time.

TABLE OF CONTENTS

Abstract

CHAPTER 1. INTRODUCTION, LITERATURE SURVEY, AND SCOPE OF DISSERTATION 1

1.1 Literature Survey 2

1.2 Differential Game Structure 9

1.3 Differential Game Formulation 14

1.4 Objective and Scope of Dissertation 17

CHAPTER 2. DEVELOPMENT OF NUMERICAL ALGORITHM 19

2.1 Differential Game Solution 20

2.2 The State of the Art on Numerical Solution 29

2.3 Differential Dynamic Programming with State and Control Constraints 35

2.3.1 Derivation of DDP with State and Control Constraints 35

2.3.2 DDP Computational Procedure 43

2.3.3 Step-Size Adjustment 44

2.4 Gradient Projection for Outer Optimization 47

2.4.1 Gradient Calculation 47

2.4.2 Gradient Projection 50

Page

2.5 Algorithm Steps........................... 53

CHAPTER 3. LINEAR QUADRATIC INTERCEPT PROBLEM.... 57

3.1 Formulation of Linear-Quadratic Differential Game........................... 57

3.2 Analytical Closed-Loop Solution............. 59

3.2.1 With Assumption that Saddle Point Exist........................... 59

3.2.2 Without Assumption that Saddle Point Exist........................... 61

3.2.3 Summary of Analytical Solutions and Discussion........................... 67

3.3 An Illustrative Example Without Control Constraint........................... 69

3.4 Linear Quadratic Problem with Hard Limit on Controls........................... 75

3.5 Numerical Solutions........................... 77

3.5.1 Algorithm Mechanization.............. 78

3.5.2 Effects of Parameter Variations...... 84

3.5.3 Discussion on the Algorithms......... 87

CHAPTER 4. A NONLINEAR STOCHASTIC PURSUIT EVASION PROBLEM........................... 94

4.1 Description of the Problem.................. 95

4.2 Formulation of the Problem 99

4.2.1 Dynamics of the Problem.............. 99

4.2.2 Cost Function of the Problem......... 106

4.2.3 Constraints.......................... 109

4.3 Convergence Control Technique.............. 110

		Page
4.4	Computational Aspects of the Problem........	114
	4.4.1 Parameter Value Assignment...........	115
	4.4.2 Maxmin Solution......................	117
	4.4.3 Minmax Solution......................	132
	4.4.4 Net Solution.........................	142
4.5	Discussion of the Problem..................	145
CHAPTER 5.	CONCLUSIONS AND RECOMMENDATIONS.......	149
5.1	Conclusions................................	149
5.2	Recommendations for Future Research.........	150

CHAPTER 1

INTRODUCTION, LITERATURE SURVEY, AND SCOPE OF DISSERTATION

Pursuit and Evasion problems have been treated by many authors as differential games. Analytically, only linear quadratic differential games have been solved. Functional Analysis has served as a good tidy approach to gain valuable insights to some aspects of differential games theory. However, only the simplest mathematical problems which represent very small or no resemblance of physical realization of real life has been solved by this approach.

Presently, the hope to solve for an optimal or near optimal solution of a *realistic pursuit and evasion* differential game does seem to lie on efficient numerical algorithms. To make this dissertation as self contained as possible, we shall start off with a brief background and history of game theory through literature survey of the game theory in general, narrow down to the work done on numerical solutions which will be included in the next chapter. A general structure of differential game will then be formulated. The formulation of mathematical model of differential game will be discussed. Lastly, we shall conclude this chapter with the statements and the significance of what we hope to accomplish in this dissertation.

1.1 Literature Survey

The problem of pursuit as a mathematical model was originated in the fifteenth century by Leonardo da Vinci according to Davis[1]. In 1732 Bougner proposed and solved for an optimal curve by which a vessel moves in pursuing another which flees along a straight line, supposing that the velocities of the two vessels are always in the same ratio. More recently, Hathaway, Archibald, and Manning,[3],[4] in 1921 worked on a more difficult problem in which the evader moves on a circle.

During the same year (1921) Emile Borel attempted to abstract strategic situations of game theory into a mathe-atical theory of strategy. After John von Neumann proved the Minimax Theorem in 1928, the theory was firmly esta-blished. However, the academic interests in the game theory did not catch on until the publication in 1944 of the impressive work by John von Neumann and Oskar Morgen-stern, Theory of Games and Economic Behavior. The theme of this book pointed out a new approach to the general problem of competetive behavior specifically in economics through a study of games of strategy. It was soon realized that the applications of the theory are not limited only to economics but also could be applied to the military, poli-tics, and other civil organizations as well.

Since then a great amount of research on game theory was published, a bibliography compiled in 1959[13] contains more than one thousand entries. It is therefore impossible to mention all these reports. Only a brief overview of the section of the field that is closely related to this dissertation will be presented here.

It is interesting to note that the games of pursuit mentioned so far in the preceeding paragraphs are one-sided optimal control problems where only the pursuers have freedom of movement while the evaders move on pre determined trajectories. A new dimension in which both players have the freedom to choose their motions was added by Isaacs when he began the development of the theory of differential games at the Rand Corporation(6). Isaacs compiled all his results in a book(7) published in 1965. Ho(8) provided the control engineers with the review of Isaacs' book in a more familiar terminology. It was here that the elements of game theory was married to the theory of optimal control. Briefly, Isaacs is concerned with problems with payoff function.

$$J(\underline{x}_0,\underline{u}(t),\underline{v}(t)) = F(\underline{x}(T),T) + \int_0^T L(\underline{x}(t),\underline{u}(t),\underline{v}(t);t)dt \tag{1.1}$$

and dynamics

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},\underline{v};t) \qquad \underline{x}(0) = \underline{x}_0 \tag{1.2}$$

where T is the final time or the time when the state trajectory meets a given terminal manifold. He assumes that a saddle point exist, an assumption which is not always true. The precise me... g a saddle point will be given in the next chapter when we discuss the solution of Differential Games. At the saddle point, the payoff function is called the value of the game and is designated by $J^*(\underline{x},t)$. Isaacs uses what he called the Tenet of Transition, a game theory equivalent of Bellman's Principle of Optimality which he apparently found independently in fact may have predated it to show that the value function must satisfy his Main Equation One, or ME_1,

$$\frac{\partial J^*}{\partial t} + \min_{\underline{u}} \max_{\underline{v}} [J_{\underline{x}}^{*T} \cdot \underline{f}(\underline{x},\underline{u},\underline{v};t) + L(\underline{x},\underline{u},\underline{v};t)] = 0 \qquad (1.3)$$

In principle, ME_1 can be used to solve for $\underline{u}^* = \underline{u}^*(x,J_{\underline{x}}^*;t)$ and $\underline{v}^* = v^*(\underline{x},J_{\underline{x}}^*;t)$. u^* and v^* are then substitued back into ME_1 to give the Main Equation Two, or ME_2,

$$\frac{\partial J^*}{\partial t} + J_{\underline{x}}^{*T} \cdot \underline{f}(\underline{x},\underline{u}^*,\underline{v}^*;t) + L(\underline{x},\underline{u}^*,\underline{v}^*;t) = 0 \qquad (1.4)$$

This is a Hamilton-Jacobi type equation and is often referred to as a Hamilton-Jacobi-Bellman equation or a pre-Hamiltonian equation which is somewhat of an injustice to Isaacs. These equations will be used in our development of a numerical algorithm in the next chapter.

Isaacs also contributes towards the sufficiency part of the solution of the game through his so called Verification Theorem. In essence, he states that if $J^*(\underline{x},t)$ is a unique continuous function satisfying the main equations and the boundary condition $J^*(\underline{x}(T),T) = F(\underline{x}(T),T)$, then J^* is the value of the game and any $\underline{u}^*$ and $\underline{v}^*$ which satisfy ME_2 and caused the desired end points to be reached are optimal. He proves this theory as the limit of a convergent series of discrete approximations to the differential game.

Gadzhiev[15] worked out necessary and sufficient conditions for the existence of a pure strategy solution for a problem with quadratic cost function and linear dynamic systems. He stated also that a pure a strategy solution for general differential game might not exist. In exploring the application of the celebrated Minimax Principle, he met only limited success because of the difficulty indefining a probability measure for the controls available for play which are time functions with infinite variability in magnitude.

The most rigorous treatment to date contain in the work of Freidman[14] and Berkovitz[10]. Friedman in his book published in 1971 uses Functional Analysis approach and went through a mathematical maze of complications to obtain essentially the same results as Isaacs. Berkovitz extended results of the classical calculus of variations to zero-sum-two-person differential games. His main results are: under same fairly restrictive conditions with the Hamiltonian-like function

$$H(\underline{x},\underline{u},\underline{v},\underline{p}) = L(\underline{x},\underline{u},\underline{v}) + \underline{p}^T.\underline{f}(\underline{x},\underline{u},\underline{v}) \qquad (1.5)$$

the optimal control u^* and v^* satisfy the following equations

$$\dot{\underline{x}} = H_{\underline{p}}(\underline{x}, \underline{u}^*, \underline{v}^*, \underline{p})$$

$$\dot{\underline{p}} = - H_{\underline{x}}(\underline{x}, \underline{u}^*, \underline{v}^*, \underline{p}) \qquad (1.6)$$

$$H_{\underline{u}} + g_{1_{\underline{u}}}^T \cdot \underline{\mu} = 0 \qquad H_{\underline{v}} + g_{2_{\underline{u}}}^T \cdot \underline{\bar{\mu}} = 0$$

$$\underline{\mu} \leq 0 \quad \mu_i g_{1i} = 0 \qquad \underline{\bar{\mu}} \geq 0 \quad \bar{\mu}_i g_{2_i} = 0$$

where g_1 and g_2 are contraint functions on $\underline{u}$ and $\underline{v}$ respectively, $\underline{\mu}$ and $\underline{\bar{\mu}}$ are associated Lagrange multipliers. He also establishes sufficiency conditions using field concepts. All these results applies under the assumption of existence of a saddle point, again we may emphasize, an assumption that is not always true.

As mentioned before in the opening statement, analytical results have indeed been rare except for the problem with linear dynamics and quadratic cost. Athans[16] presents a review of recent works on differential games. Ho, Bryson and Baron[17] model and solve a persuit-evasion problem as a linear quadratic game, deriving conditions for existence of solution. The meagerness in analytical results according to Ho[8] is a direct consequence of the complications and the complexities introduced into the optimal control problem by the "other" controller.

McFarland[18] stated that most authors elect to treat each player's control with no constraint using integral penalities in the cost function to preclude any solution with infinite magnitude. Published results have indeed been rare for differential games with bounded control. Progress were made by Meschler[19] and Mier[20] on deterministic problems of simple construction permitting analytical treatment. Mier suggested that under close examination generalization cannot be made. Other authors have made some headways in this respect using numerical analysis. These will be mentioned in the next chapter.

Another interesting approach to differential games is the so called geometric approach. Some of the more significant contributions in this respect are the work of Blaguiere, Gerard, and Leitman[21],[22],[23] in an augmented state space. Karlin and Shapley[24] also used geometric approach to provoke a rigorous investigation into the geometry of moment spaces. The more recent works an geometric approach to game theory are those of Westphal[25] and Westphal and Stubberud[26] where they synthesize mixed strategies and find game values for both sclar and vector controls. Herrelko[27] later extended these results to cover the case with information time lag.

Although many questions still remained unanswered for two-person zero-sum dynamic games with perfect information and pure strategies, many researchers have wandered into the area of other games. One reason for this is because the early works were not applicable to many real-world problems which are often n-person, non-zero sum and stoachastic. Each of these areas is a challenge in itself and most of the efforts to date have been rightly concentrated on each area individually.

Analytical success with the linear-quadratic problem has induced many authors to explore stochastic games. Most of the works in this area have been on two person-zero sum linear-quadratic games with noisy transitions of dynamics, random initial conditions, or moisy observations. According to Bryson and Ho[27] the main effort has been to relate solutions of these problems to the "certainty-equivalence" principle of stochastic optimal control theory. This, however, contain a logical fallacy in the treatment either implicitly or explicitly of one player's control in his opponent's estimator. Some of the contributors in the area of stochastic differential games are Ho, Speyer, Behn and Ho, Rhodes and Luenberger, Willman, Mons, Bley, etc.

To conclude this very brief overview of the historical aspects of differential games, it might be worthwhile to mention that successful researchers have shown respect for this quite new field, and realize that the complications involved is far more than an extension of optimal control. Progress is made in careful steps and examples are kept simple so that the new concepts being uncovered can be made clear.

1.2 Differential Game Structure

In this section, an informal presentation of a very general type of differential game, where there are any number of players with different cost criteria and different information sets will be given. With this structure, some general classifications of differential game will be made. Figure 1 illustrates basic structures of a general differential game. The interval of play is assumed to be [o,T] where T may be a fixed final time or the time when the state trajectory first reach a given terminal manifold.

At each instantanous time t in the interval [o,T], each player from the total number of N players chooses a vector of control inputs, u_i, to optimize his cost criteria:

$$J_i(u_1,\ldots,u_N;t) = F_i(x(T),T) + \int_0^T L_i(x,u_1,\ldots,u_N;t)\,dt \quad (1.7)$$

$i = 1,2,\ldots, N$

These controls serves as input vectors to a common dynamic system(shared by all players) described by a nonlinear vector differential equation:

$$\dot{x} = f(x,u_1, \ldots,u_N,t,w(t))$$

$$x(0) = x_o \quad (1.8)$$

Where w(t) is a vector input of random noise usually

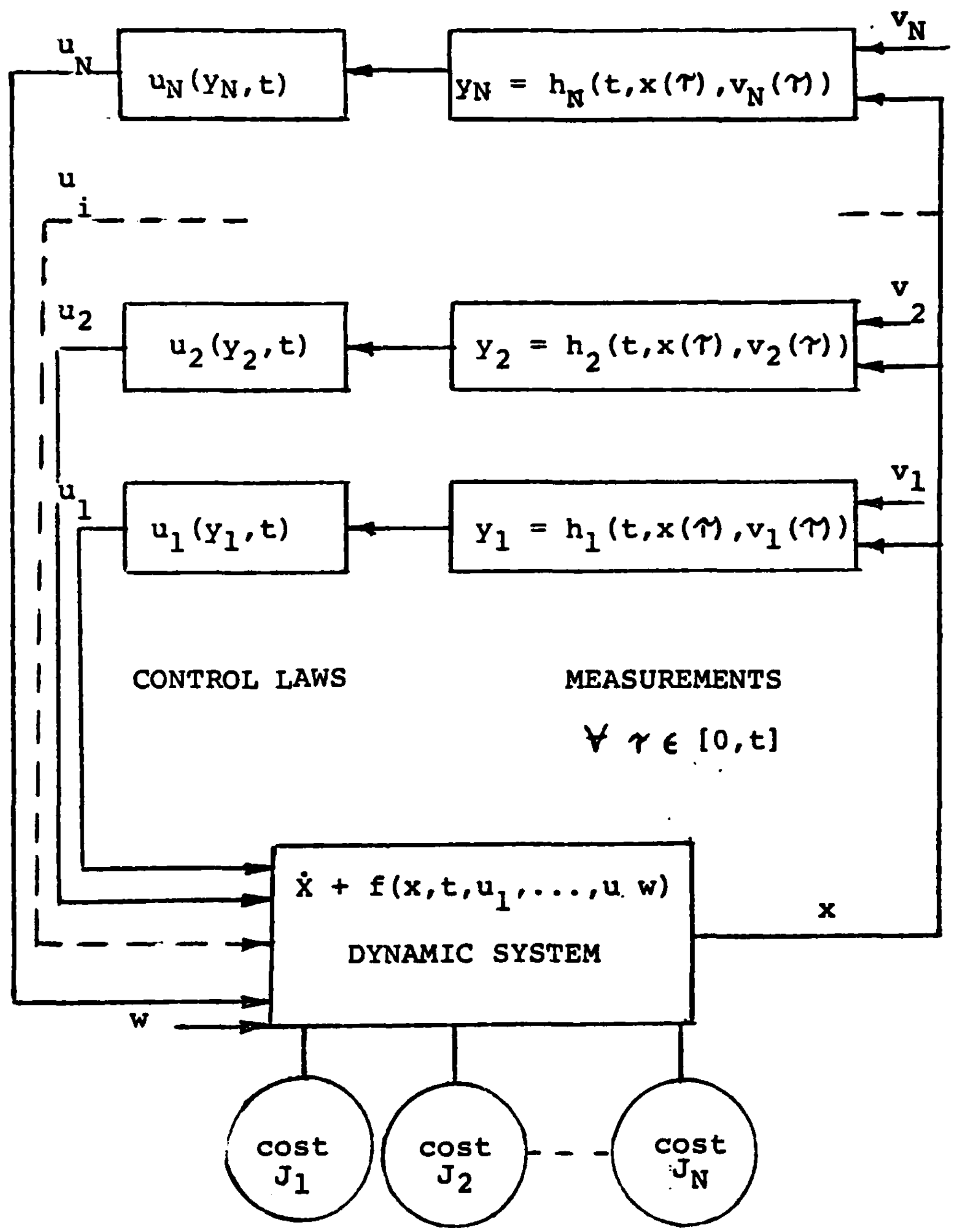


Figure 1. Structure of a general differential game

Gaussian.

Generally some contraints $u_i \in U_i$ where

$$U_i = [u_i : g(u_1, \ldots, u_N \cdot t) \leqslant 0] \qquad (1.9)$$

or a set of vector contraint equations is also imposed on the choice of control vectors u_i's.

At each particular time t, each player also has a set of measurements or information sets available to aid his decision in choosing the control vector These information sets are accumulated by each player during the interval [0,t] in the form

$$y_i = h_i(t, x(\tau), v_i(\tau)) \text{ for all } \tau \in [0,t] \qquad (1.10)$$

where $u_i(\tau)$ is the noise vector input to each of the player measurement system.

To date, most differential games are formulated in two special cases as follow:

(1) $h_i(t, x(\tau), v_i(\tau) = x(\tau) \quad \text{for } 0 \leqslant \tau \leqslant t$

where we have a deterministic system or perfect measurements of the state vector if all information is used, the solution is in <u>closed-loop</u> form.

$$(2)\quad h_i(t,x(\tau),v_i(\tau) = \begin{cases} x_o & \text{for } t = 0 \\ 0 & \text{for } t > 0 \end{cases}$$

then only the initial state vector is known the system is still deterministic, but in this case the solution can only be generated in open-loop form. It is interesting to note here that even if perfect measurements are available, the controller may still not be able to generate a closed-loop solution depending on the relative sizes of the computation time and the duration of the game. There are only a few simple cases namely with linear dynamics and quadratic pay-offs where closed-loop solution can be generated in closed-form. Other more difficult cases which recently have drawn some interests from researchers are:

$$(3)\quad h_i(t,x(\tau),v_i(\tau)) = H_i(\tau)x(\tau) + v_i(\tau)$$

$$\text{for } 0 \leqslant \tau \leqslant t$$

where $H_i(\tau)$ may be either time-invariant or time-variant matrix. $v_i(\tau)$ is additive white Gaussian noise. In this case we have stochastic differential game with linear measurements. Again only linear-quadratic differential games have been solved with this information set.

$$(4) \quad h_i(t, x(\tau), v_i(\tau)) = \begin{cases} x_o & \text{for } t = 0 \\ 0 & \text{for } t > 0 \end{cases}$$

where x_o is a random-variable usually Gaussian. Here, we also have a stochastic game. A non-linear version of pursuit-evasion differential game with this set of information will be presented later-on in this report.

$$(5) \quad h_i(t, x(\tau), v_i(\tau)) = x(\tau) \quad \text{for } 0 \leq \tau \leq t - \sigma$$

here we have perfect measurements with time delay. Only a handful of papers about differential games were generated on this set of information. All the authors limited themselves to simple problems in this case because of the tremendous complexities involved.

Once the measurement is made and one of the information set from those listed above is formed, a control law can then be generated. In deterministic cases, control laws are generated directly either from analytical or numerical solutions. In linear quadratic stochastic cases, it is well known that the"Certainty Equivalence Principle" or the "Separation Theorem" can be extended from the theory of stochastic optimal control. That is, the estimation process can be carried out first using some type of filter, the most well known being Kalman Bucy's, then the estimated states can be used to generate a control law as if they were deterministic.

Furthur classification of differential game can be made from the cost functions. If $\sum_{i=1}^{N} J_i = 0$ then the game is called the N -person Zero sum differential game If $\sum_{i=1}^{N} J_i \neq 0$ then it is called N-person non-zero sum differential game. One important class is the two-person zero-sum differential game. This is the class of differential game that we shall be concerned with throughout this dissertation. An exact formulation of two-person zero-sum differential game will be given in the next section.

1.3 Differential Game Formulation

From hereon in this report, we shall be concerned with two-person zero-sum differential games, whose formulation is given as follows:

There are two opposing players U and V who choose their control strategies to drive the equation of motion of the dynamic system:

$$\frac{d}{dt}\, \underline{x}(t) = \underline{f}[\underline{x}(t), \underline{u}(t), \underline{v}(t), t]; \; \underline{x}(0) = \underline{x}_o \qquad (1.11)$$

where

$\underline{x}(t)$ = state vector of dimension n x 1

$\underline{u}(t)$ = control input of the minimizing player, dimension m x 1

$\underline{v}(t)$ = control input of maximizing player, dimension p x 1

The duration of play [0,T] is fixed, the terminal time T is given explicitly in the problem. The vector function $\underline{f}$ is assumed piecewise continuous during the interval of play, and differentiable up to any order required in all its arguments.

The vector control functions $\underline{u}(t)$ and $\underline{v}(t)$ are piecewise continuous, differentiable function of time, and belong to some prescribed feasibility regions $\underline{u}(t) \in U$ and $\underline{v}(t) \in V$ where

$$U = [\underline{u}: \ \underline{g}_1(\underline{x},\underline{u},t) \leq 0] \tag{1.12}$$

$$V = [\underline{v}: \ \underline{g}_2(\underline{x},\underline{v},t) \leq 0] \tag{1.13}$$

$\underline{g}_1$ and $\underline{g}_2$ are vector state and control contraints, dimension $\leq$ m and p respectively

U is minimizing and V is maximiaing the following scalar cost functional:

$$J(\underline{u},\underline{v}) = F(x(T)) + \int_0^T L(\underline{x}(t),\underline{u}(t),\ v(t),t)dt \tag{1.14}$$

The scalar functions F and L are also continuous and differentiable up to any order required in their arguments.

The feasible regions U and V generally preclude the use of infinite controls by either player. They usually impose hard limits on the control vectors. Equation (1.11) implies that both players agree on the dynamic structure of the system. In reality this is always an approximation. The practicability of the solution then depends on how close one can model (1.11) to represent the real system. This is not surprising, however, since in all mathematical modelling of real physical problem, some simplified assumptions generally have to be made to ensure mathematical and computational tractability of the model of the problem.

Note that generally the terminal time T does not have to be given explicitly. Instead a terminal manifold of the form $\underline{\psi}(x(T),T) = 0$ can be described. This, however, cannot ensure the termination of the game. Therefore, the fixed duration game which can be considered as a special case of terminal manifold where $\underline{\psi}(x(T),T) = T-t_f = 0$ is chosen here to eliminate any termination problem that may arises in order that other important concepts can be investigated and clarified.

It is also interesting to note that if hard limits are not imposed on the controls in (1.12) and (1.13), then additional assumptions will have to be made on the controls to ensure that the magnitude of the optimizing controls will be finite. This is done in McFarland[18]. Briefly, he defines regions of finite controls C^v and C^u such that whenever U try to use infinite control in minimizing then V can choose $\underline{v}(t) \in C^v$ to drive the cost functional very large. Similarly, whenever V try to use infinite control in maximizing, then U can choose $\underline{u}(t) \in C^u$ to obtain very small value for the cost functional. These assumptions are not required in this report.

1.4 Objective and Scope of Dissertation

As mentioned before on the opening statements of this chapter, an efficient algorithm is needed before a solution of pursuit and evasion differential game can be implemented effectively. Most of the results of the computational methods developed so far have been found under the assumption that the saddle point exists. A discussion of this point and a counter example will be presented in the next chapter. McFarland[18] worked out an algorithm without assuming existence of a saddle point. He uses Differential Dynamic Programming from the work of Jacobson an Mayne[30] for inner optimization and gradient method for outer optimization. McFarland results, however, does not contain any hard limit on any control or state constraint as in (1.12) and (1.13). Our work then will be as follow:

1.4.1. Using an approach similar to McFarland's, an algorithm will be developed to handle hard limits on control and state variables of differential games. The Differential Dynamic Programming used in the inner optimizations will be modified to handle the constraints. Some gradient projection schemes will have to be used to cope with the outer optimization. This will be presented in chapter 2.

1.4.2 A linear quadratic pursuit and evasion differential game will be investigated. The case without any control constraint will be solved analytically. Through a simple illustrative example, physical outcomes corresponding to parameters of the problem will be investigated. The case with control constraint cannot be solved analytically. Two numerical solutions will be offered, one using the algorithm developed in 1.4.1 and another using an indirect approach with the assumption of existence of the saddle point and direct application of differential Dynamic Programming similar to the algorithm used by Neeland[31] and later by Jarmark[32],[33],[34].
Any similarity or discrepancy, advantage and disadvantage between the two methods will be reported. Chapter 3 will cover this.

1.4.3 Chapter 4 will cover a stochastic nonlinear model for a missle-anti missle intercept problem. A mathematical model will be developed using a set of sufficient statistics as state variables. The problem will then be solved using the algorithm developed in chapter 2.

1.4.4 Chapter 5 will summarized all the results accumulated in this report. Recommendations for future research will be presented.

CHAPTER 2

DEVELOPMENT OF NUMERICAL ALGORITHM

It is widely accepted that for a general differential game, a numerical solution is generally needed. In such case, therefore, only open-loops form of solution can be generated. However, if numerical algorithm can be developed with such simplicity that the total computation and implementation time is much less than the duration of the game, the process can be repeated with either the determined or the approximated new states treated as initial states depending upon whether the problem is deterministic or stochastic. In this manner, a closed-loop solution can be approached as a limit as the computation and implementation times get smaller and smaller.

On the other hand, one must be careful that in trying to simplify the problem, assumptions are not made that pertinent physical realizations must be sacrified. Thus the control engineer must strife to seek the delicate balance between these two points. This is an optimization problem in itself. The solutions which will be presented in this report will not be claimed as optimal in this sense but they will be developed with these two points in mind.

Before we actually start off with the development of the numerical algorithm, it would seem appropriate to discuss the meaning of the solution of differential game to get a clear picture of what we are looking for. The state of the art on numerical solution can then be surveyed to pave our way towards the solution. The actual algorithm will be composed of two parts: the inner optimization using Differential Dynamic Programming with state and control constraints, and the outer optimization using gradient projection. Finally, we shall conclude this chapter with the details of the steps of the algorithm.

2.1 Differential Game Solution

In game theory, the solution for each player is the choice of the strategy that he has to choose among many possible ones. In choosing his strategy, a player cannot be sure about the outcome of the game because he does not have any "a priori" knowledge about his opponent's choice. This is the fact that caused more complications in the theory of differential games than just being simply an extension of optimal control theory.

In two person zero-sum differential game, the players are two adversaries confronting one another with one's loss being the other's gain. Therefore, each player will try to minimize the maximal loss his opponent can caused. The strategy that realizes this outcome becomes his solution. Once the solution is found, the player does not care what strategy his opponent will use. He is that much better off if his opponent does not use the strategy that caused him the maximal loss. it is for this reason that some authors have called this type of solution the security level of the game.

In the following discussion, let us designate the two opponents as follows:

minimizing player = U, using control = $\underline{u}(t)$

maximizing player = V, using control = $\underline{v}(t)$

t is any instant between the interval [0,T]

the cost functional of the game, $J(\underline{u},\underline{v})$ is in the form of equation (1.14)

First, let us look at the minimizing player's, U, point of view. For any arbitrary control $\underline{u}(t)$ that he chooses, he is assured that at maximal the cost will be

$$J(\underline{u},\hat{\underline{v}}) = \max_{\underline{v}(t)} J(\underline{u},\underline{v}) \tag{2.1}$$

Naturally, U will choose the control $\hat{\underline{u}}(t)$ which will minimize the maximum cost

$$J(\hat{\underline{u}},\hat{\underline{v}}) = \min_{\underline{u}(t)} [\max_{\underline{v}(t)} J(\underline{u},\underline{v})] \tag{2.2}$$

Thus, the solution for U is the so called minnimax solution, $\hat{\underline{u}}(t)$. Note that U does not care whether V will use $\hat{\underline{v}}(t)$ or not because from equation (2.2) we can see that

$$J(\hat{\underline{u}},\underline{v}) \leqslant J(\hat{\underline{u}},\hat{\underline{v}}) \quad \text{for } v \neq \hat{v} \tag{2.3}$$

The equality in equation (2.3) is included because of the possibility of the non-uniqueness of $\hat{\underline{v}}$. Therefore, U will usually gain if V uses any other control besides $\hat{\underline{v}}$.

Now, from V's point of view, for any arbitary control $\underline{v}(t)$

$$J(\tilde{\underline{u}},\underline{v}) = \min_{\underline{u}(t)} J(\underline{u},\underline{v}) \tag{2.4}$$

V is assured that his cost will be at least $J(\tilde{\underline{u}},\underline{v})$. Since his objective is to maximize, V will choose $\tilde{\underline{v}}(t)$ that will maximize this minimum cost

$$J(\tilde{\underline{u}},\tilde{\underline{v}}) = \max_{\underline{v}(t)} [\min_{\underline{u}(t)} J(\underline{u},\underline{v})] \tag{2.5}$$

Thus, $\tilde{\underline{v}}$ is the <u>max min</u> solution for V. Again V does not care if U uses $\tilde{\underline{u}}$ or not. From (2.5) it is clear that V will almost always gain and at least will not loose if U uses any other control than $\tilde{\underline{u}}$ because

$$J(\underline{u},\tilde{\underline{v}}) \geqslant J(\tilde{\underline{u}},\tilde{\underline{v}}) \text{ for } \underline{u} \neq \tilde{\underline{u}} \tag{2.6}$$

The <u>net cost</u> of the game is $J(\hat{\underline{u}},\tilde{\underline{v}})$. Generally, we can state that each player will usually benefit from using the <u>secure</u> strategies as depicted in the following equation

$$J(\tilde{\underline{u}},\tilde{\underline{v}}) \leqslant J(\hat{\underline{u}},\tilde{\underline{v}}) \leqslant J(\hat{\underline{u}},\hat{\underline{v}}) \tag{2.7}$$

From a general viewpoint, there is absolutely no reason to presume that

$$\hat{\underline{u}} = \tilde{\underline{u}} \text{ and } \hat{\underline{v}} = \tilde{\underline{v}} \tag{2.8}$$

If (2.8) is true, however, $\hat{\underline{u}}$ and $\tilde{\underline{v}}$ will constitue a saddle point for the game, and the net cost, $J(\hat{\underline{u}},\tilde{\underline{v}})$, is called the value of the game.

Definition 1: The differential game described in section 1.3 is said to have a value if

$$\min_{\underline{u}(t)} \left[\max_{\underline{v}(t)} J(\underline{u},\underline{v})\right] = \max_{\underline{v}(t)} \left[\min_{\underline{u}(t)} J(\underline{u},\underline{v})\right] \tag{2.9}$$

where $\underline{u}$ ranges over U and $\underline{v}$ ranges over V.

Definition 2: If a game has the value J^*, and if there exists $(\underline{u}^*,\underline{v}^*)$ such that $J^* = J(\underline{u}^*,\underline{v}^*)$ and

$$J(\underline{u}^*,\underline{v}) \leqslant J(\underline{u}^*,\underline{v}^*) \leqslant J(\underline{u},\underline{v}^*) \tag{2.10}$$

then $\underline{u}^*$ is optimal for U and $\underline{v}^*$ is optimal for V. The pair $(\underline{u}^*,\underline{v}^*)$ is called a saddle point. $\underline{u}^*$ and $\underline{v}^*$ are called pure strategy solution.

Most of the previous works on differential game have been concentrated on pure strategy solution, and the conditions for which it exists. However, for a general nonlinear nonquadratic problem a saddle point does not generally exist.

Two examples will be shown here to demonstrate this point. The first one is dued to McFarland[18] concerning a single stage static game. The second one is dued to Berkovitz[35] concerning differential game with nonlinear dynamic.

Example 1: Let the controls be scalars u and v and the cost be the polynomial function of u and v as follows:

$$J(u,v) = (u^4-2u^2+2)\left(-\frac{v^4}{18} + \frac{v^2}{18} + \frac{2}{3}\right) + (u^3-3u)\left(\frac{v^3}{18}-\frac{2v}{9}\right) \qquad (2.11)$$

The cost function is formed in such a way that neither U nor V can use infinite control in their optimization process. Using previous terminology $c^v = [v: |v| < 2]$ and $c^u = [u: u \epsilon R]$ where R is any number on the real line. The solutions to this *problem are:*

For player U, Minmax: $\hat{u} = \pm 1, \; (\hat{v} = \pm 1)$

$$J(\hat{u},\hat{v}) = 1$$

For player V, Maxmin: $\tilde{v} = 0, \; (\tilde{u} = \pm 1)$

$$J(\tilde{u},\tilde{v}) = \frac{2}{3}$$

Net cost of the game: $J(\hat{u},\tilde{v}) = \frac{2}{3}$

It is interesting to note that McFarland called the points

(+1,+1) and (-1,-1) local saddle points. These points, however, are not saddle points according to the definition given above. Using definition 2, it is obvious that the saddle point does not exist in this problem. This difference occurred because McFarland defined a saddle point as the point where the gradients of the cost with respect to controls simultaneously vanish accompanied by some simple second order condition. In this report, the name saddle point will be preserved for such point when the control pairs of the minmax and the maxmin solutions are the same and pure strategy exists. McFarland also worked out the solutions of this example in details which will not be repeated here.

Example 2: For a game of fixed final time $T > 0$, play terminates at $t = T$. The cost function being minimized by U and maximized by V is given by

$$J(u,v) = \int_0^T x\,dt \tag{2.12}$$

The state x is determined by the dynamic equation and the initial condition

$$\dot{x} = (v - u)^2, \quad x(0) = x_o \tag{2.15}$$

The controls are constrained by $u = U(t,x)$, where U is piecewise continuous differentiable on the interval $[0,T]$

and $0 \leqslant U(t,x) \leqslant 1$, and $v = V(t,x)$, where V is also piecewise continuous differentiable on the interval $[0,T]$ and $0 \leqslant V(t,x) \leqslant 1$.

Maxmin Solution:

For any arbitrary control chosen by V, U can choose the same strategy and thus guarantee that

$\dot{x} = 0$ on the whole interval $[0,T]$

Therefore, $\max_{v} \min_{u} J(u,v) \leqslant x_o T$

For any pair (u,v), however, it is obvious that $\dot{x} \geqslant 0$.

Thus,

$$J(u,v) = \int_0^T x\,dt \geqslant x_o T$$

Therefore, $\max_{v} \min_{u} J(u,v) = x_o T$ (2.14)

Minmax Solution:

For any arbitrary control chosen by U, V can practically guarantee that $\dot{x} \geqslant 1/4$ on $[0,T]$ by choosing his strategy as follows:

$$v = \begin{cases} 1 & \text{if} \quad u \leqslant 1/2 \\ 0 & \text{if} \quad u > 1/2 \end{cases}$$

using this strategy V can make $x(t) \geqslant x_o + \frac{t}{4}$

Hence, $\min_{u} \max_{v} J(u,v) \geqslant x_o T + \frac{T}{8}$ (2.15)

Now, if U choose $u \equiv 1/2 \ \forall [0,T]$, then for any v, $\dot{x} \leqslant 1/4$ on $[0,T]$.

Thus, upon integrating we have

$$\min_{u} \max_{v} J(u,v) \leqslant x_o T + \frac{T^2}{8} \tag{2.16}$$

From (2.15) and (2.16) we can conclude that

$$\min_{u} \max_{v} J(u,v) = x_o T + \frac{T^2}{8} \tag{2.17}$$

In summary, we have $J(\tilde{u},\tilde{v}) = x_o T$

$$J(\hat{u},\hat{v}) = x_o T + \frac{T^2}{8}$$

Therefore, for this problem

$$J(\tilde{u},\tilde{v}) < J(\hat{u},\hat{v})$$

Again, a saddle point does not exist in this game, and the game does not have a value in pure strategies.

From these examples, we see that even for simple games (the first is a static game and the second even though has nonlinear dynamic contain linear cost) saddle point does not have to exist. Sufficiency conditions for saddle points were worked out by many authors, but they are restricted to a very limited class of differential game.

One question arises on what then is the true solution of differential game in the case where a saddle point does not exist. The celebrated Minimax Principle of game theory asserts that in this case the players can find fixed probability laws from which random strategy (among those possible) can be selected in such a way that the average value of the cost sustained by each player comprises the value of the game in the long run. The probability laws that have to be found is contained in the mixed-strategy solution. The main disadvantage for this type of solution is that it is not only hard to implement but also exceedingly complicated if not impossible to solve for in a realistic pursuit-evasion differential game. All the researchers who worked on mixed-strategy had to resort to very simple problems which bear little or no physical significance.

At the present, we should be contented with the security level type of solution. The minmax and the maxmin solutions provide the least-maximum-loss strategy for each player. If the player can accept the cost accrued from using his least-maximum-loss strategy, then he can rest assured that he will not be worse off no matter what strategy his oppenent will use. One critical argument against this type of solution is that it is too conservative. However, in view of the fact that numerical solutions are needed for all realistic pursuit-evasion differential games, all the strategies implemented will be suboptimal to some extent. The less complicated the solution can be the closer it will be to a true optimal. In addition to the computation and the implementation time involved, this should more than outweighed any advantage that could be gained by using the mixed-strategy solution. This report then will be aimed at finding an efficient algorithm to solve for the least-maximum-loss strategies or the minmax and the maxnim solutions without the assumption of existence of a saddle-point.

2.2 The State of the Art on Numerical Solution

As mentioned before in section 1.1, Berkovitz[10] used calculus of variation approach to form a set of equations for two person zero-sum differential game. Blaquiere[22] also has a similar development but the emphasis there is put in the geometric aspect of the game. These works have become the basic fundamentals for most numerical methods developed thereafter. Most authors require the assumption of the existence of a saddle-point to provide the pure-strategy solution. The main feature for these techniques is having to solve a two-point boundary value problem (TPBVP). This type of problem is encountered very frequently in optimal control theory and in mathematics, they involved a set of differential equations with initial conditions given on some variables and final conditions given on the rest. Since the optimization process using this approach does not involve evaluation of the cost function directly in each iteration, it has been labeled the indirect methods. Bryson and Ho[28] suggested that numerical methods for TPBVP can be cagegorized into three methods: gradients, quasilinearization (Newton-Raphson), and neighbouring extremal.

Recently, Jacobson and Mayne[30] has added a very efficient new technique to solve the optimal control problem using Differential Dynamic Programming (DDP). This method differs to the above indirect methods in that rather than having to solve the TPBVP, a set of associated equations are derived with all the final values given. The task of integrating backward is much simpler than the task of solving the TPBVP. Moreover, the convergence time of DDP is generally found to be more rapid than any of the three methods mentioned above.

All three indirect methods mentioned have some common features. Each method start off with some nominal solution for which <u>some</u> boundary conditions are satisfied, and each use informations provided by a linearized solution about the nominal solution to improve the solution for successive iteration until <u>all</u> the boundary conditions are satisfied. The rate of convergence differs greatly as they are applied to various problems. Generally, the gradient method exhibits a fast convergence to start off but becomes relatively poor near the optimal. Some phenomenal such as zig-zagging has been known to be closely associated with this method near the optimal value. Newton-Raphson or quasilinearization converges quadratically near optimal but the initial guess must be chosen very carefully. To this end neighbouring extremal is generally even more sensitive to the initial guess.

All gradients methods exhibit one common difficulty namely the so called "step-size" problem. That is, after a feasible direction is found using the gradient, how far should the control correction be applied in that direction. Too small a step-size will cause a drastic decrease in convergence rate whereas too big a step-size sometimes leads to non-convergence. There are two basic techniques to take care of this problem. The first one was devised by Jacobson an Mayne(30) and used effectively by McFarland(18) in his dissertation. This has to do with adjusting the time interval $[\tau, T]$ on which the new control is found in such a way that the variation of the states is not too large. The second technique was introduced by Jarmark(32),(33),(34) where the quadratic terms of the controls are added to the integral terms of the cost functional and the weighting matrices are chosen in such a way that again the variation in the states is acceptable. Both techniques have exhibited very good convergence property.

Leffler(36) developed theortically a numerical algorithm containing two phases. The first is called the "gradient" phase in which the directions of the control changes are computed, and the second is called the "restoration" phase which is needed to keep the new control within the feasible region. Theoretically, Leffler's algorithm is capable of handling constraints on both states and controls. Computationally, however, the pursuit-evasion problem that he solved does not include any significant constraint on either the states or control inputs of each players.

On the application aspects, Robert and Montgomery[37] attempted to solve a classical pursuit-evasion problem. The distance between the two aircrafts at the time of closest approach was used as the cost index. They were successful in obtaining the optimal trajectories for most initial conditions. They also found some regions where the trajectories are not unique and remedied the situation by adding the time until interception term into the cost function. The dynamics were non-linear and the controls were subjected to hard constraints. The computation time required, however, was very large. Approximately the computation time was ten times as great as the engagement time in their report.

The most complex air-to-air combat model so far documented was worked out by Lynch[38]. The emphasis was to formulate and solve as realistic a mathematical model of air-to-air combat as possible. A three dimensional model was used. All the involved factors were considered. Thrust, drag, and lift were stored as a monlinear function of altitude and airspeed. The controls were roll-angle, thrust, and turn rate with the latter two subjected to hard oonstraints. The cost index used is the time required for the pussuer to manuver closer to the evader than some given radius. Again Lynch used the gradient method with the same step-size adjustments as Robert and Montgomery to obtain satisfactory convergence for most initial conditions. He also reported on singular surfaces where non-unique solutions were encountered. Needless to say the computional time needed were horrendous. Roughly the requirement for the computational time is one hundred times in magnitude as compared to the simulated encounter time.

Leatham[39] also studied the same model mentioned above using the method of "neighbouring optimal trajectories". This method is closely associated to the "successive sweep" method of Dyer and McReynolds.[40] The details will not be described here, interest readers can refer to the above references. It might be worthwhile to mention, however, that even if the missing initial conditions can be accurately guessed, the computational time for this method took roughly twenty times greater than the time of engagement. Dethlefsen[41] performed an analytical analysis of neighbouring optimal method for a much simpler problem. No numerical result, however, was included in that report.

Graham[42] entended the quasilinearzation technique of optimal control to cover the first -order necessary conditions he derived for differential game. The technique was then used to solve a pursuit-evasion game involving a ground-to-air intercepter and a supersonic airplane essentially the same unconstrained problem solved by Leffler. This method is very sensitive to the choice of the initial trajectory. The magnitude of the computational time is approximately ten times that of the encountered time.

Neeland[31] used Differential Dynamic Programming (DDP) to develop algorithm to solve a realistic air-to-air combat game under the assumption that the saddle point exists. Even though the development of the algorithm contains the term up to second-order. The actual algorithm used to solve the pursuit-evasion problem is actually a first order algorithm which is practically required because he was looking for a very fast computation time. He actually reduced the computational time to be smaller than the engagement time in the non-sigular case. Jarmark also confirmed this for a large number of sample problems. Therefore, we must conclude that of all the techniques available so far, Differential Dynamic Programming is the most efficient one. Details development of the first order algorithm of DDP will be included in the next section.

All the methods discussed so far have been under the assumption of existence of a saddle point. Reports with no a-priori assumption of a saddle point have been rare indeed. McFarland worked out one such report. Besides having no assumption of a saddle point, his technique differs from the indirect method in that the evaluation of the cost function is required in each iteration. Therefore, McFarland's technique is sometime referred to as a direct method or a direct solution technique. Briefly for an arbitrary control the inner optimization of this method is performed by using second order DDP to locate all the local maxima (minima) created by the opponent's control. The player's control is then adjusted by using either the "steepest decent" or the "conjucate gradient" methods. This adjustment is called the outer-optimization. The process is then reiterated. Since McFarland does not consider any control or state constraint, the termination criteria occurs when the variation of the Hamiltonian with respect to the player's control is negligible in which case a saddle point is located or when a cross-over point is located in which case the solution will not be a saddle-point. The exact definition of a cross-over point will be given in the following section.

2.3 Differential Dynamic Programming with State and Control Constraints

We shall start our development with the inner optimization process. Even though the actual derivation is for a maximin solution, it can also be applied to a minmax solution simply by the substitution of control variables and the interchange between the minimization and the maximization within the procedure.

2.3.1. Derivation of DDP with State and Control Constraints

For a maxmin solution, with any arbitrary control $\underline{v}(t)$ chosen by V, the differential game formulated in section 1.3 becomes a constrained optimal control problem as follows:

Player U now chooses his control strategy to drive the equation of motion of the dynamic system:

$$\frac{d}{dt}\underline{x}(t) = \underline{f}(\underline{x}(t),\underline{u}(t),t); \ \underline{x}(0) = x_o \tag{2.18}$$

$t \in [0,T] \ = \ \theta$, T is fixed

$\underline{x}(t) \in X$ an n dimensional Euclidean space

$\underline{u}(t) \in U$

$$U = [u\colon \theta \rightarrow R^m, \ \underline{g}(\underline{x},\underline{u},t) \leq 0] \tag{2.19}$$

where R^m is an m dimensional Euclidean space, the mapping is bounded by the constraints vector hyperplane $\underline{g}(\underline{x},\underline{u},t) \leq 0$ of dimension $\leq m$

U is trying to minimize the following cost function

$$J(u(t)) = \int_o^T L(\underline{x},\underline{u},t)dt + F(x(T),T) \tag{2.20}$$

Since the starting time is arbitrary, we can rewrite this cost function using the imbedding principle

$$J(\underline{x}(t),t,\underline{u}(\tau)) = F(\underline{x}(T),T) + \int_t^T L(\underline{x}(\tau),\underline{u},(\tau),\tau)d\tau \tag{2.21}$$

We then define

$$J^*(\underline{x}(t),t) = \min_{\underline{u}(\tau)} J(\underline{x}(t),t,\underline{u}(\tau))$$

$$= \min_{\underline{u}(\tau)} [F(\underline{x},(T),T) + \int_t^T L(\underline{x}(\tau),\underline{u}(\tau),\tau)d\tau] \tag{2.22}$$

By using the well known principle of optimality in optimal control theory (2.22) can be rewritten as follows:

$$J^*(\underline{x}(t),t) = \min_{\underline{u}(\tau)} [F(\underline{x}(T),T) + \int_t^{t+\Delta t} L\, d\tau + \int_{t+\Delta t}^{T} L\, d\tau] \quad (2.23)$$

the arguments of the functionals L in (2.23) is the same as those in equation (2.22). Combining the first and last term in the bracket and using the above definition we have

$$J^*(\underline{x}(t),t) = \min_{\underline{u}(\tau)} [J^*(\underline{x}(t+\Delta t),t+\Delta t) + \int_t^{t+\Delta t} L\, d\tau] \quad (2.24)$$

Expand $J^*(\underline{x}(t+\Delta t),t+\Delta t)$ in Taylor's series about $(x(t),t)$ for small Δt,

$$0 = \min_{\underline{u}(t)} [\frac{\partial J^*}{\partial t} \cdot \Delta t + \frac{\partial J^*}{\partial \underline{x}} \cdot \underline{f}(\underline{x}(t),\underline{u}(t),t) \cdot \Delta t + L(\underline{x}(t),\underline{u}(t),t)$$

$$\cdot \Delta t + o(\Delta t)] \quad (2.25)$$

where $\frac{o(\Delta t)}{\Delta t} \rightarrow 0$ as $t \rightarrow 0$

Dividing (2.25) throughout by Δt and let Δt approaches zero, we have

$$\frac{\partial J^*}{\partial t} + \min_{\underline{u}(t)} [L(\underline{x}(t),\underline{u}(t),t) + \frac{\partial J^{*T}}{\partial \underline{x}} \cdot \underline{f}(\underline{x}(t),\underline{u}(t),t)] = 0 \quad (2.26)$$

the partial derivatives are evaluated at the point $(x(t),t)$ This is the well known Bellman's equation in the optimal control theory and serves as a starting point for DDP. Define the Hamiltonian as

$$H(\underline{x},\underline{u}, J^*_{\underline{x}},t) = L(\underline{x},\underline{u},t) + J^{*T}_{\underline{x}} \cdot \underline{f}(\underline{x},\underline{u},t) \tag{2.27}$$

$$\text{where } J^*_{\underline{x}} \equiv \frac{\partial J^*}{\partial \underline{x}} \equiv \left[\frac{\partial J^*}{\partial x_1} \quad \frac{\partial J^*}{\partial x_2} \cdots\cdots \frac{\partial J^*}{\partial x_n}\right]^T \tag{2.27}$$

the superscript T stands for transpose. (2.26) then becomes

$$J^*_t(\underline{x}^*(t),t) + \min_{\underline{u}(t)} H(\underline{x}^*(t),\underline{u}(t), J^*_{\underline{x}},t) = 0 \tag{2.28}$$

Since we are dealing with the state and control constraints, a penalty term can be incorporated into the cost functional as follows

$$J(\underline{x}(t),\underline{u}(t),t) = F(x(T),T) + \int_0^T [L(\underline{x},\underline{u},t) + \underline{\mu}^T \cdot \underline{g}(\underline{x},\underline{u},t) \; dt \tag{2.29}$$

With constraints then Bellman equation becomes

$$J^*_t(\underline{x}^*(t),t) + \min_{\underline{u}\in U} [H(\underline{x}^*(t),\underline{u}(t), J^*_{\underline{x}},t) + \underline{\mu}^T \cdot \underline{g}(\underline{x}(t),\underline{u}(t))] \tag{2.30}$$

where $\underline{\mu}$ is the Lagrange multiplier vector and can be solved for by using the Khun-Tucker condition from non-linear programming. This is included in the appendix A.

For now, it suffices to say that, the vector multiplier function, $\underline{\mu}(\underline{x},\underline{u},t)$ is identically equal to zero when the corresponding constraint is a strict <u>inequality</u>. Otherwise.

$$\underline{\mu} = -[\bar{g}_{\underline{u}} \cdot \bar{g}_{\underline{u}}^T]^{-1} \cdot \bar{g}_{\underline{u}} \cdot H_{\underline{u}} \tag{2.31}$$

where $^{-}$ designates the boundary in U or when the constraint is a strict equality.

It is a well known fact that except for a few simple cases, Bellman equation cannot be solved analytically. DDP provides an excellent iteration procedure for numerical solutions. We now proceed to derive pertinent equations required for DDP.

Considered equation (2.30), if there exists a nonimal $\bar{\underline{x}}$ such that $\underline{x}^*(t) = \bar{\underline{x}}(t) + \Delta\underline{x}(t)$ where $\Delta\underline{x}(t)$ is small in the interval [0,T] then

$$J_t^*(\bar{\underline{x}} + \Delta\underline{x},t) + \min_{\underline{u}\in U} [H(\bar{\underline{x}} + \Delta\underline{x},\underline{u},J_{\underline{x}}^*,t) + \underline{\mu}^T(\bar{\underline{x}}+ \Delta\underline{x},\underline{u},t) \cdot \underline{g}(\bar{\underline{x}}+ \Delta\underline{x},\underline{u},t)] = 0 \quad (2.32)$$

Expand (2.32) in Taylor's series to first order in $\Delta\underline{x}$ about $\bar{\underline{x}}$ and using the so called complimentary slackness in the Khun-Tucker condition we get

$$\frac{\partial}{\partial t} J^*(\bar{\underline{x}},t) + \frac{\partial}{\partial t} J_{\underline{x}}^{*T}(\bar{\underline{x}},t) \cdot \Delta\underline{x} + \min_{\underline{u}\in U} [H(\bar{\underline{x}},\underline{u},J_{\underline{x}}^*,t)$$

$$+ H_{\underline{x}}^T(\bar{\underline{x}},\underline{u},J_{\underline{x}}^*,t) \cdot \Delta\underline{x} + (J_{\underline{xx}}^* \cdot \underline{f}(\bar{\underline{x}},\underline{u},t))^T \cdot \Delta\underline{x}$$

$$+ \underline{\mu}^T(\bar{\underline{x}},\underline{u},t) \cdot \underline{g}_{\underline{x}}(\bar{\underline{x}},\underline{u},t) \cdot \Delta\underline{x} + \text{h.o.t.} = 0 \quad (2.33)$$

h.o.t. stands for higher order terms in the Taylor's series expansion.

If $\Delta \underline{x}$ is very small, as $\Delta \underline{x}$ approaches zero, the higher order terms also tend towards zero, and we can split (2.33) into two parts as follows

$$\frac{\partial}{\partial t} J^*(\underline{\bar{x}},t) + H(\underline{\bar{x}},\underline{u}^*,J^*_{\underline{x}},t) = 0 \qquad (2.34a)$$

$$\frac{\partial}{\partial t} J^*_{\underline{x}}(\underline{\bar{x}},t) + H_{\underline{x}}(\underline{\bar{x}},\underline{u}^*,J^*_{\underline{x}},t) + J^*_{\underline{xx}} \cdot f(\underline{\bar{x}},u^*,t)$$

$$+ \underline{g}^T_x(\underline{\bar{x}},\underline{u}^*,t) \cdot \underline{\mu}(\underline{\bar{x}},\underline{u}^*,t) = 0 \qquad (2.34b)$$

where $\underline{u}^* = \min_{\underline{u} \in U}^{-1} H(\underline{x},\underline{u},J^*_{\underline{x}},T)$ or in words $\underline{u}^*$ is the feasible control with which the Hamiltonian is minimized.

Since $J^*(\underline{x}(t),t)$ and $J^*_{\underline{x}}(\underline{x},(t),t)$ are functions of $\underline{x}(t)$ and t, the total derivatives with respect to t are

$$\frac{d}{dt} J^*(\underline{\bar{x}},t) = \frac{\partial}{\partial t} J^*_{\underline{x}}(\underline{\bar{x}},t) + J^{*T}_{\underline{x}}(\underline{\bar{x}},t) \cdot \underline{f}(\underline{\bar{x}},\underline{\bar{u}},t) \qquad (2.35a)$$

$$\frac{d}{dt} J^*_{\underline{x}}(\underline{\bar{x}},t) = \frac{\partial}{\partial t} J^*_{\underline{x}}(\underline{\bar{x}},t) + J^*_{\underline{xx}}(\underline{x},t) \cdot \underline{f}(\underline{\bar{x}},\underline{\bar{u}},t) \qquad (2.35b)$$

We now define an estimates cost change at time t as

$$a(t) = J^*(\underline{\bar{x}},t) - \bar{J}(\underline{\bar{x}},t) \qquad (2.36)$$

where $\bar{J}(\underline{\bar{x}},t)$ is the nominal cost that occured when U is using the control strategy $\underline{\bar{u}}(t)$. Note that

$$\frac{d}{dt} \bar{J}(\underline{\bar{x}},t) = -L(\underline{\bar{x}},\underline{\bar{u}},t) \qquad (2.37)$$

Substituting (2.35a) and (2.37) into the total time derivative of definition (2.36) we have

$$\dot{a}(t) = \frac{\partial}{\partial t} J^*(\underline{\bar{x}},t) + J^*_{\underline{x}}(\underline{\bar{x}},t)\cdot\underline{f}(\underline{\bar{x}},\underline{\bar{u}},t) + L(\underline{\bar{x}},\underline{\bar{u}},t) \tag{2.38}$$

Notice that the last two terms on the right hand side of (2.38) is the definition of a Hamiltonian. Using (2.34a) for $\frac{\partial}{\partial t} J^*(\underline{\bar{x}},t)$ then (2.38) becomes

$$-\dot{a}(t) = H(\underline{\bar{x}},\underline{u}^*,J^*_{\underline{x}},t) - H(\underline{\bar{x}},\underline{\bar{u}},J^*_{\underline{x}},t) \tag{2.39}$$

Substituting (2.34b) into the negative of (2.35b) we obtain

$$\begin{aligned} -\dot{J}^*_{\underline{x}}(\underline{\bar{x}},t) = {} & H_{\underline{x}}(\underline{\bar{x}},\underline{u}^*,J^*_{\underline{x}},t) + \underline{g}^T_{\underline{x}}(\underline{\bar{x}},\underline{u}^*,t)\cdot\underline{\mu}(\underline{\bar{x}},\underline{u}^*,t) \\ & + J^*_{\underline{xx}}\cdot[\underline{f}(\underline{\bar{x}},\underline{u}^*,t) - \underline{f}(\underline{\bar{x}},\underline{\bar{u}},t)] \end{aligned} \tag{2.40}$$

Consider the last term on equation (2.40)

$$J^*_{\underline{xx}}\cdot[\underline{f}(\underline{\bar{x}},\underline{u}^*,t) - \underline{f}(\underline{\bar{x}},\underline{\bar{u}},t)] \approx J^*_{\underline{xx}}\cdot\underline{f}_{\underline{u}}(\underline{\bar{x}},\underline{\bar{u}},t)\cdot\Delta\underline{u} \tag{2.41}$$

Expand the dynamic equation (2.18) about $\underline{\bar{x}},\underline{\bar{u}}$ to the first order, we get a differential equation describing an approximation of the deviation in $\underline{x}$ as follows

$$\frac{d}{dt}\Delta\underline{x} = \underline{f}_{\underline{x}}(\underline{\bar{x}},\underline{\bar{u}},t)\cdot\Delta\underline{x} + \underline{f}_{\underline{u}}(\underline{\bar{x}},\underline{\bar{u}},t)\cdot\Delta\underline{u} \tag{2.42}$$

$$\Delta \underline{x}(0) = 0 \tag{2.42}$$

Throughout the derivation process, we are depending on the assumption that $\Delta \underline{x}$ is small. Therefore, (2.42) can be used to show that $\underline{f}_{\underline{u}}(\overline{\underline{x}},\overline{\underline{u}},t) \cdot \Delta \underline{u}$ will also be small for small $\Delta \underline{x}$. This suggests that the term (2.41) can be neglected. Neglecting the $J^*_{\underline{xx}}$ terms in (2.40) introduces an error

$\Delta J^*_{\underline{x}}(t)$ in $J^*_{\underline{x}}(t)$ of order

$$\int_T^t |\underline{u}^*(t_1) - \underline{u}(t_1)| \, dt \tag{2.43}$$

The integration is performed backward because the final conditions of differential equations (2.40) and (2.41) are given as we shall see later. We define

$$|\underline{u}| = \sum_{i=1}^{m} |u_i|$$

The error $\Delta a(t)$ in $a(t)$ is of order

$$\int_T^t \int_T^{t_2} |\underline{u}^*(t_1) - \overline{\underline{u}}(t_1)| \, dt_1 |\underline{u}^*(t_2) - \overline{\underline{u}}(t_2)| \, dt_2 \tag{2.44}$$

From equation (2.39) it is clear that $a(t)$ is of order

$$\int_T^t |\underline{u}^*(t_3) - \overline{\underline{u}}(t_3)| \, dt_3 \tag{2.45}$$

From (2.44) and (2.45), we can see that if either $T - t$ is of order ε or $|\underline{u}^* - \overline{\underline{u}}|$ is of order ε , then the error $\Delta a(t)$ is of order ε^2 while a(t) itself is of order ε . Thus, we have shown that by neglecting the term (2.41), a(t) will still be correct to the first order.

At $t = T$; $J^*(\overline{\underline{x}}(T),T) = F(\overline{\underline{x}},(T),T)$ (2.46)

hence we have the following final conditions

$$a(T) = 0 \tag{2.47}$$

$$J^*_{\underline{x}}(\overline{\underline{X}}(T),T) = F_{\underline{x}}(\overline{\underline{x}}(T),T) \tag{2.48}$$

In summary, the equations that constitue the heart of DDP in the case where the state and the control constraints are present are

$$-\dot{a}(t) = H(\overline{\underline{x}},\underline{u}^*,J_{\underline{x}},t) - H(\overline{\underline{x}},\overline{\underline{u}},J_{\underline{x}},t); \quad a(T) = 0$$

$$-J^*_{\underline{x}}(\overline{\underline{x}},t) = H_{\underline{x}}(\overline{\underline{x}},\underline{u}^*,J^*_{\underline{x}},t) + g^T_{\underline{x}}(\overline{\underline{x}},\underline{u}^*,t) \cdot \mu(\overline{\underline{x}},\underline{u}^*,T); \tag{2.49}$$

$$J^*_{\underline{x}}(\overline{\underline{x}}(T),T) = F_{\underline{x}}(\overline{\underline{x}},(T),T)$$

2.3.2 DDP Computational Procedure

(1) Use a nominal control $\overline{\underline{u}}(t)$, integrate (2.18) forward to obtain a nominal trajectory $\overline{\underline{x}}(t)$. Store $\overline{\underline{x}}(t),\overline{\underline{u}}(t)$ together with the computed cost $\overline{J}(\overline{\underline{x}},t)$ from (2.20).

(2) Integrate (2.49) backward from T to 0 while simultaneously minimizing $H(\overline{\underline{x}},\underline{u},J^*_{\underline{x}},t)$ with respect to $\underline{u}(t)$

$\in U$ to get $\underline{u}^*(t)$. Store $\underline{u}^*(t)$ and $a(t)$.

(3) Again integrate (2.18) forward using $\underline{u}^*(t)$ and also compute $J^*(\underline{x}^*,t)$. If the actual cost change agrees with the predicted cost change computed in (2) then $\underline{u}^*(t)$ can be accepted as a new nominal control.

(4) If $|a(0)| < \epsilon$ where ϵ is some small positive number, then $\underline{u}^*(t)$ is regarded as optimal control. If not then steps (1),(2) and (3) are reiterated.

(5) If the actual cost change differs too much from the predicted cost change, then the "step-size" method can be applied.

2.3.3 Step-Size Adjustment

Substituting the minimizing control in each iteration, $\underline{u}^*(t)$ into (2.18), we obtain

$$\frac{d}{dt}(\bar{\underline{x}} + \Delta\underline{x}) = \underline{f}(\bar{\underline{x}} + \Delta\underline{x},\underline{u}^*,t);\ \bar{\underline{x}}(0)+\Delta\underline{x}(0) = x_o \qquad (2.50)$$

Because $\Delta\underline{x}(0) = 0$, the size of $\Delta\underline{x}(t)$ is dued to the variation in control $\Delta\underline{u} = \underline{u}^* - \bar{\underline{u}}$ as can be seen from equation (2.50). One way to restrict the size of $\underline{x}(t)$ is to restrict the interval of time over which (2.50) is integrated. This is desirable since we do not wish to alter the size of $\underline{u}^*(t)$ found in the minimization process of $H(\bar{\underline{x}},\underline{u},J^*_{\underline{x}},T)$ in step (2) of the algorithm.

Throughout the derivation process of the DDP equation (2.49), we were under the assumption that $\Delta\underline{x}$ is small.

This is an important assumption, because if $\Delta \underline{x}$ is not small enough, then the higher order terms in (2.33) will not be negligible. This in turn caused the actual cost change, ΔJ, to deviate too much from the predicted cost change, a.

Let $0 \leqslant t_I \leqslant T$, use nominal control $\bar{\underline{u}}(t)$ from $0 \leqslant t \leqslant t_I$, then $\underline{x}(t_I) = \bar{\underline{x}}(t_I)$ and $\Delta \underline{x}(t)=0$ from $0 \leqslant t \leqslant t_I$. Now, use the minimization control $\underline{u}^*(t)$ to integrate (2.50) forward from t_I to T. If $[t_I, T]$ is small, then $\underline{x}$ will be small for finite $\underline{u}$. Note that

$$a(\bar{\underline{x}}, t_I) = \int_T^{t_I} [H(\bar{\underline{x}}, u^*, J^*_{\underline{x}}, t) - H(\bar{\underline{x}}, \bar{\underline{u}}, J^*_{\underline{x}}, t)]dt \qquad (2.57)$$

One criteria to determine whether the actual cost change "agrees" with the predicted cost change is as follow:

$$\frac{\Delta J}{a(\bar{\underline{x}}, t_I)} > C; \qquad C \geqslant 0 \qquad (2.52)$$

There is no strict rule to govern the size of C. It is up to the judgement of the control engineer to decide what positive number he should use for C for his particular problem. Usually C is set around 0.5. It might be noted that C should be less than 1 since the actual cost change should not exeed the predicted cost change. This, however, could happen.

If (2.52) is satisfied, then $\Delta \underline{x}$ is small enough, and the iteration process is repeated using the minimizing control as a new nominal control in the interval $[t_I, T]$. Usually t_I is originally set at 0. If (2.52) is not satisfied, then set $t_I = \frac{T}{2}$ and repeat. If the criteria is still not satisfied using the new t_I, then use $t'_I = \frac{T-t_I}{2} + t_I$ as the new starting point of the interval for integration. The process is repeated until the criteria is satisfied. Figure 2 is used to illustrate this scheme.

$$0 \text{————————————————} t_I \text{————————} t'_I \text{———} t''_I \text{———} T$$

Figure 2. Half interval scheme to control the size of $\Delta \underline{x}$

Following are a few characteristics of the "step size adjustment":

(1) The new nominal trajectory may have a corner at t_I because $\underline{\bar{u}}(t_I)$ may be different from $\underline{u}^*(t_I)$. The integration routine used, therefore, must be capable of handling differential equations with discontinuous controls.

(2) If the minimizing trajectory coincides with the nominal trajectory during the latter portion of the interval but the nominal trajectory is non-optimal in the earlier portion, two steps must be taken: First, a(t) must be monitered while performing the backward integration in step (2) of the routine, note the time $t = t_{eff}$ when $a(t) > 0$ or when a(t) is equal to or greater than a small positive

constant. Then $[0,t_{eff}]$ can be used instead of $[0,T]$.

(3) Since computer is used in the routine, the quantized requirement may eventually conflict with the repetition in halving of the interval $[0,T]$, i.e., $[t_I,T]$ may eventually be smaller than one quantized step. This difficulty may be remedied by either adjusting C or use a smaller quantization for the integration.

2.4 Gradient Projection for Outer Optimization

Two basic ideas are covered in this section. First we must see how the function-space gradient of the minimum cost with respect to variations in the maximizing control $\underline{v}(t)$ can be computed. Second the fact that the search direction must lead us to a new feasible point and yet increases the minimum cost as much as possible must be considered.

2.4.1 Gradient Calculation

We are given $J(u^*(v),v)$.

To find the change in J due to a variation in v, we have

$$\frac{dJ}{d\underline{v}} = \frac{\partial J}{\partial \underline{v}} + \frac{du^*}{d\underline{v}} \cdot \frac{\partial J}{\partial \underline{u}}$$

If the minimum obtained from the inner optimization process is not on the boundary, then $\frac{\partial J}{\partial \underline{u}}$ evaluated at the extremal would be equal to zero. In this case then the gradient of J with respect to a variation in $\underline{v}$ would be qual to $\frac{\partial J}{\partial \underline{v}}$.

In general, the first-order necessary conditions that $\underline{u}^*(t)$ minimizes $J(\underline{u},\underline{v})$ for any given $\underline{v}(t)$ are

$$J^*(\underline{u}^*,\underline{v}) = F(\underline{x}^*(T)) + \int_0^T L(\underline{x}^*(t),\underline{u}^*(t),\underline{v}(t),t)dt \qquad (2.53)$$

$$\dot{\underline{x}}^*(t) = \underline{f}(\underline{x}^*(t),\underline{u}^*(t),\ \underline{v}(t),t);\quad x^*(0) = \underline{x}_o \qquad (2.53b)$$

$$\dot{J}^*_{\underline{x}} = -H_{\underline{x}};\quad J^*_{\underline{x}}(T) = F_{\underline{x}}(\underline{x}^*(T),T) \qquad (2.53c)$$

$$H_{\underline{u}} + \underline{\mu}^T \cdot \underline{g}_{\underline{u}} = 0 \qquad (2.53d)$$

all the partial derivatives are evaluated at $(\underline{x}^*(t), J^*_{\underline{x}}, \underline{u}^*(t),\underline{v}(t))$.

Consider equation (2.53d), we anticipate that the previously optimal quantities $\underline{u}^*(t),\underline{x}^*(t),J^*, J^*_{\underline{x}}(t)$ will have some variations with a small variation in the given $\underline{v}(t)$ designated $\Delta\underline{v}(t)$. We shall call these variations $\Delta\underline{u}^*(t)$, $\Delta\underline{x}^*(t)$, ΔJ^*, and $\Delta J_{\underline{x}}(t)$ respectively. Expanding (2.53d) to first order and subtracting out all nominal quantities, we get

$$(H_{\underline{ux}} + \underline{\mu}^T \cdot \underline{g}_{\underline{ux}}) \cdot \Delta\underline{x}^* + (H_{\underline{uu}} + \underline{\mu}^T \cdot \underline{g}_{\underline{uu}}) \cdot \Delta\underline{u}^*$$

$$+ (H_{\underline{uv}} + \underline{\mu}^T \cdot \underline{g}_{\underline{uv}}) \cdot \Delta\underline{v} + \underline{f}_{\underline{u}} \cdot \Delta J^*_{\underline{x}} = 0 \qquad (2.54)$$

where care must be taken in the definition of the above partial derivatives to ensure that all the matrix and tensor operations in equation (2.54) are compatible.

Moreover $\Delta J^*_{\underline{x}}$ can be approximated to the first order by $J^*_{\underline{xx}} \cdot \Delta \underline{x}^*$, equation (2.54) then becomes

$$(H_{\underline{ux}} + \underline{\mu}^T \cdot \underline{g}_{\underline{ux}} + \underline{f}_{\underline{u}} \cdot J^*_{\underline{xx}}) \cdot \Delta \underline{x}^* + (H_{\underline{uu}} + \underline{\mu}^T \cdot \underline{g}_{\underline{uu}}) \cdot \Delta \underline{u}^*$$

$$+ (H_{\underline{uv}} + \underline{\mu}^T \cdot \underline{g}_{\underline{uv}}) \cdot \Delta \underline{v} = 0 \tag{2.55}$$

Solving (2.55) for the change in $\underline{u}^*$ with respect to a small variation in $\underline{v}$ we obtain

$$\frac{d\underline{u}^*}{d\underline{v}} = (H_{\underline{uu}} + \underline{\mu}^T \cdot \underline{g}_{\underline{uu}})^{-1} \cdot [(H_{\underline{ux}} + \underline{\mu}^T \cdot \underline{g}_{\underline{ux}} + \underline{f}_{\underline{u}} \cdot J^*_{\underline{xx}}) \frac{d\underline{x}^*}{d\underline{v}}$$

$$+ H_{\underline{uv}} + \underline{\mu}^T \cdot \underline{g}_{\underline{uv}}] \tag{2.56}$$

All the terms on the right hand side of equation (2.56) contain second order partial derivatives. An error analysis similar to the one given in section 2.3.1 can be given to show that $\frac{d\underline{u}^*}{d\underline{v}}$ will be of second order as compared to the variation $\Delta \underline{v}$ and ΔJ. Since we are dealing with a first order calculation, $\frac{d\underline{u}^*}{d\underline{v}}$ can be neglected. In general then the gradient of J with respect to a variation in $\underline{v}$ approximated to the first order would be equal to $\frac{\partial J}{\partial \underline{v}}$. It is well known from calculus of variation that at any instant t, $\frac{\partial J}{\partial \underline{v}}$ is equal to $H_{\underline{v}}$.

2.4.2 Gradient Projection

We are now dealing with the problem

$$\underset{\underline{v}}{\text{Max}}\ [\underset{\underline{u}}{\text{Min}}\ J] \quad = \quad \underset{\underline{v}}{\text{Max}}\ J^*$$

$$\text{s.t.}\ \underline{g}(\underline{x},\underline{u},\underline{v},t) \leqslant 0 \tag{2.55}$$

McFarland used the gradient as the search direction for his algorithm. With the presence of the state and control constraints, however, the gradient direction may lead to an infeasible point for the next iteration. It is obvious, therefore, that some adjustments will have to be made to counter this drawback. The most well known method in nonlinear programming to handle this situation is the so called "projection method". In essence, linear constraints, or linearized constraints, form a linear manifold (defined by the region formed by the intersection of the constraints). The gradient direction can then be projected onto this manifold to produce a search direction $\underline{s}$ such that $J^{*T}_{\underline{v}} \cdot \underline{s} > 0$ so that the movement in the direction $\underline{s}$ will cause an increase in the functional J^* at a new feasible point.

Let A_q be the q x p matrix of active constraints. Actually A_q will depend on $\underline{v}$, this dependence, however, will be suppressed here to save space.

$$A_q = \begin{bmatrix} \frac{\partial g_1}{\partial v_1} & \cdots\cdots & \frac{\partial g_1}{\partial v_p} \\ \cdots & \cdots & \cdots \\ \frac{\partial g_q}{\partial v_1} & \cdots\cdots & \frac{\partial g_q}{\partial v_p} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1p} \\ \cdots & \cdots & \cdots \\ a_{q1} & \cdots & a_{qp} \end{bmatrix} \tag{2.56}$$

We note here that if the constraints are linear with respect to the controls which is usually the case in the pursuit and evasion problem, then the elements a_{ij}'s of the matrix A_q are just the eoefficients of the control variables in the active constraints.

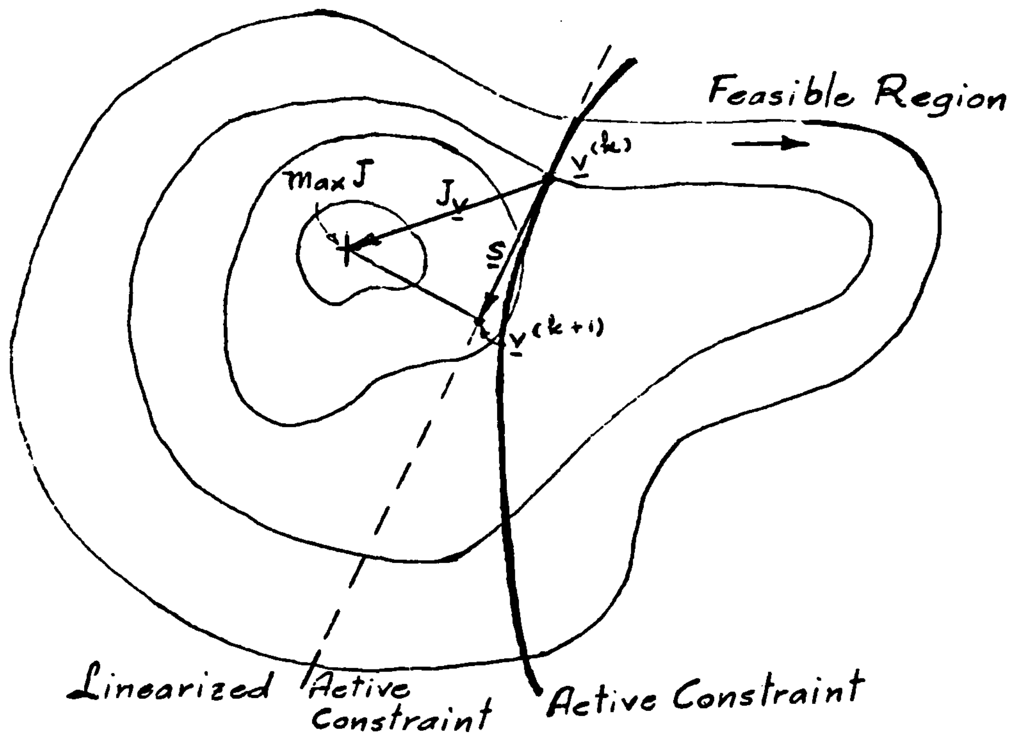


Figure 3 Gradient projection method applied at an active constraint. If the constraint is nonlinear $\underline{v}^{(k+1)}$ may be infeasible.

If the active constraints are linear, the search direction $\underline{s}$ will lie along the constraints themselves. If not, then the search direction will lie along the hyperplanes tangent to the constraints at $\underline{v}^{(k)}$ where the supercript (k) indicates the number of iteration. If $\underline{v}^{(k+1)}$ proves to be infeasible as shown in figure 3, a restoration phase will be used to move to the closest feasible point. Since the problems we shall be dealing with later on contain constraints which are linear with respect to the controls, we shall be concerned only with such problem from hereon. For this class of problem, at the end of each iteration, we shall always end up at a feasible point.

To compute the projection, let us called the linear manifold formed by the active constraints $\mathcal{M}$. Assuming regularity of the active constraints, A_q is a matrix of rank $q < p$. Since $\underline{s}$ must lie in $\mathcal{M}$ we must have $A_q \cdot \underline{s} = 0$. Using the projection theorem in functional analysis, the gradient $J^*_{\underline{u}}$ can be decomposed into two parts as follows:

$$J^*_{\underline{u}} = \underline{s} + A_q^T \cdot \underline{\beta} \tag{2.57}$$

where $\underline{s} \in \mathcal{M}$ and $A_q^T \cdot \underline{\beta} \perp \mathcal{M}$

Multiply (2.57) throughout by A_q and use the fact that $A_q \cdot \underline{s} = 0$ we have

$$A_q \cdot \underline{s} = A_q \cdot J^*_{\underline{u}} - (A_q A_q^T) \cdot \underline{\beta} = 0 \tag{2.58}$$

From which we get

$$\underline{\beta} = (A_q A_q^T)^{-1} A_q . J_{\underline{u}}^* \tag{2.59}$$

aubstitue this back into (2.58) and manipulate then

$$\underline{s} = [I - A_q^T (A_q A_q^T)^{-1} A_q] \cdot J_{\underline{u}}^* \tag{2.60}$$

The matrix

$$P = [I - A_q^T (A_q A_q A_q^T)^{-1} A_q] \tag{2.61}$$

is called the projection matrix or the projection operator on the vector $J_{\underline{u}}^*$ with respect to the subspace $\mathcal{M}$. The outer optimization terminates when $\| \underline{s} \|$ is arbitrarily small and all $\beta_i \geqslant 0$ where β_i are components of $\underline{\beta}$ computed from equation (2.59).

2.5 Algorithm Steps

As mentioned before, we shall cover the algorithm steps required only for the maxmin operation. The minmax operation is similar with the interchange in controls for the inner and the outer optimization and the appropriate change in signs in the search directions.

Starting Procedure

(1) Select a nominal control $\underline{v}(t)$, $v_o(t)$, by suitable logic using some physical insights or whatever is readily available.

(2) Calculate all the local minima of $J(\underline{u},\underline{v}_o)$ and rank them in ascending order

$J_o^{(1)}(\underline{u}_o^{(1)},\underline{v}_o)$, $J_o^{(2)}(\underline{u}^{(2)},\underline{v}_o),\ldots,J_o^{(n)}(\underline{u}^{(n)},\underline{v}_o)$

where $\underline{u}_o^{(i)}$ = locally minimizing control $i = 1,2,\ldots,n$

<u>Following steps applied for the $k^{\underline{th}}$ and $(k+1)^{\underline{th}}$ iteration ($k = 0,1,\ldots$)</u>

(3) Calculate $J_{\underline{u}_k}^{(1)}$ and its norm $\| J_{\underline{u}_k}^{(1)} \|$

$$\text{where} \quad \| J_{\underline{u}_k}^{(1)} \| = \int_o^T J_{\underline{u}_k}^{(1)T}(t) . J_{\underline{u}_k}(t)dt \tag{2.62}$$

(4) If $\| J_{\underline{u}_k}^{(1)} \| < \varepsilon_1$, a small positive constant, exit a saddle point is located at $(\underline{u}_k^{(1)},\underline{v}_k)$. If not continue.

(5) Find search direction s_k using

$$\underline{s}_k = P_k . J_{\underline{u}_k}^{(1)} \tag{2.63}$$

$$\text{where } P_k = I - A_{q_k}^T (A_{q_k} A_{q_k}^T)^{-1} A_{q_k} \tag{2.64}$$

$$\text{(6) Calculate } \| \underline{s}_k \| = \int_o^T \underline{s}_k^T(t) . \underline{s}_k(t)dt \tag{2.65}$$

$$\text{and} \quad \underline{\beta}_k = (A_{q_k} A_{q_k}^T)^{-1} A_{q_k} . J_{\underline{u}_k}^{(1)} \tag{2.66}$$

(7) If $\| \underline{s}_k \| < \varepsilon_2$, a small positive constant, and all $\beta_{k_i} \geq 0$ where β_{k_i} are elements of the multiplier vector $\underline{\beta}_k$, exit a solution is located on the boundary of the problem. If $\| \underline{s}_k \| < \varepsilon_2$ and any $\beta_i < 0$, remove that corresponding constraint and go to (5). If $\| \underline{s}_k \| > \varepsilon_2$ continue.

(8) Form new control

$$\underline{v}_{k+1}(t) = \underline{v}_k(t) + \lambda_k \underline{s}_k(t) \tag{2.67}$$

where λ_k is a suitable stepsize logic.

If $\underline{v}_{k+1}(t)$ activates another constraint, then that constraint must be added to the matrix $A_{q_{k+1}}$.

(9) Calculate all the new local minima of J_{k+1} $(\underline{u},\underline{v}_{k+1})$:

$$J^{(1)}_{k+1}(\underline{u}^{(1)}_{k+1},\underline{v}_{k+1}),\quad J^{(2)}_{k+1}(\underline{u}^{(2)}_{k+1},\underline{v}_{k+1}),\ldots\ldots J^{(n)}_{k+1}\ (\underline{u}^{(n)}_{k+1},\underline{u}_{k+1})$$

(10) If

$$J^{(i)}_{k+1}\ (\underline{u}^{(i)}_{k+1},\underline{v}_{k+1}) < J^{(1)}_{k+1}\ (\underline{u}^{(1)}_{k+1},\underline{v}_{k+1});\ i \doteq 1 \tag{2.68}$$

one of the previously higher minima has over taken the heretofore mincost. The crossover point has been overshot, go to (11). If not increase the number of iteration by one, go to (3), and reiterate.

(11) Find the cross-over point where

$$J^{(i)}_{k+1}(\underline{u}^{(i)}_{k+1},\underline{v}_{k+1}) = J^{(1)}_{k+1}\ (\underline{u}^{(1)}_{k+1},\underline{v}_{k+1}) \tag{2.69}$$

where i is the index when the inequality in (2.68) is true.

The resulting $\underline{v}_{k+1}$ that satisfies (2.69) is the required maxmin solution.

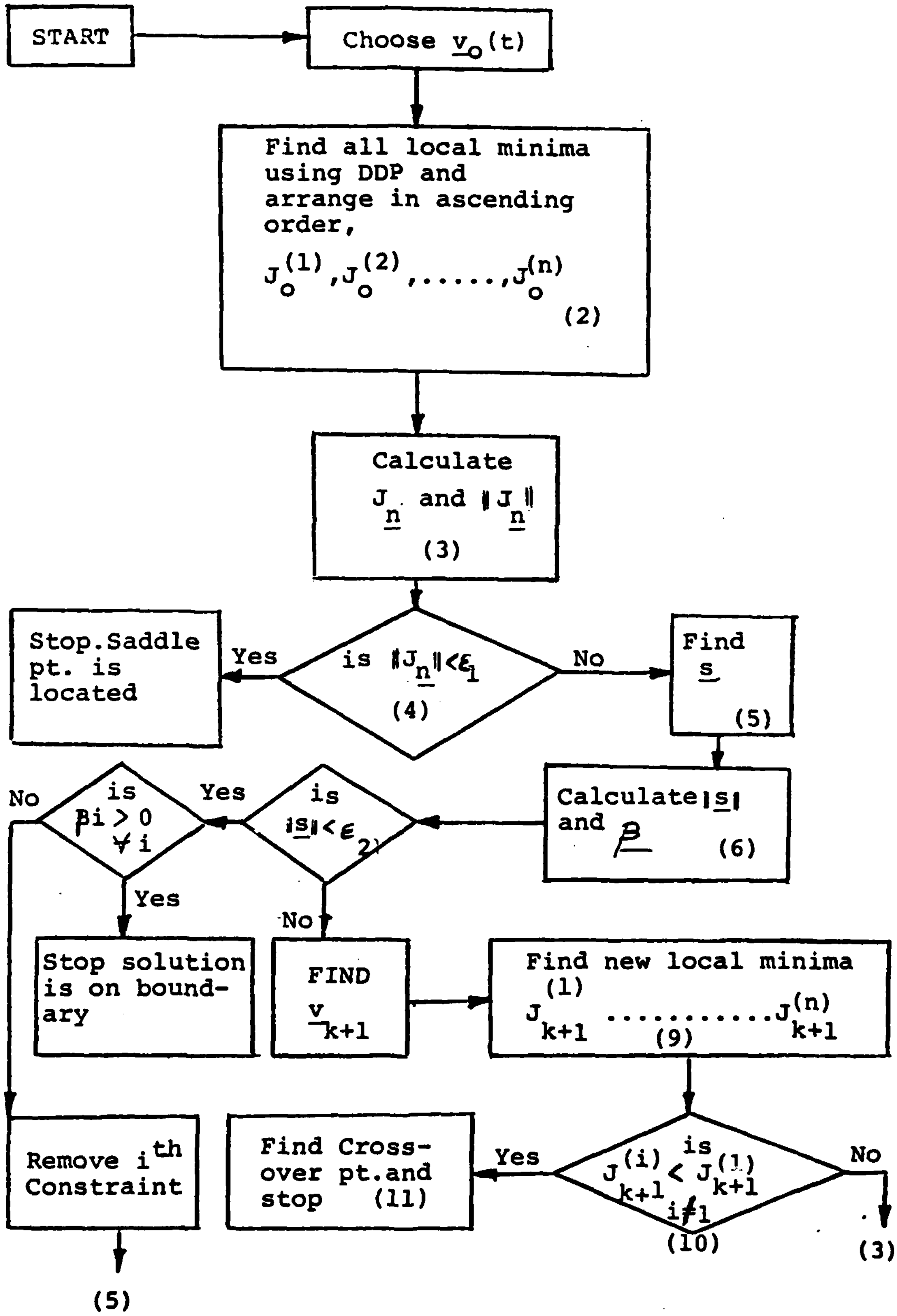


Figure 4 Flow chart of the Maxmin solution

CHAPTER 3

LINEAR QUADRATIC INTERCEPT PROBLEM

In this chapter, a general linear-quadratic differential game will be formulated and reduced into a more simple form. Analytical solutions will be presented in the non-constrained case to illustrate the difference in the level of difficulties between the cases when we have and when we do not have the assumption of the existence of a saddle point. A linear-quadratic pursuit and evasion differential game will be investigated. The case without any control constraint will be solved analytically. Through a simple illustrative example, physical outcomes corresponding to parameters of the problem will be investigated. The case with control constraints cannot be solved analytically. Two numerical solutions will be offered. In the first one an indirect approach with the assumption of existence of a saddle point and direct application of DDP similar to the algorithm used by Neeland and later by Jarmark will be used. In the second one the algorithm developed in chapter 2 will be used to solve the same problem without the assumption of a saddle point. All the problems considered in this chapter will be assumed to have perfect informations.

3.1 Formulation of Linear-Quadratic Differential Game

Generally, a linear-quadratic game will have the following cost functional:

$$J(\underline{u},\underline{v}) = 1/2\ \underline{y}^T(T)C^TC\underline{y}(T) + 1/2 \int_o^T [\underline{u}^T(t)Q_1\underline{u}(t) - \underline{v}^T(t)Q_2 v(t)]dt$$

where $\underline{y}$ = n x 1 state vector (3.1)

C = r x n terminal weighting matrix $r \leqslant n$

$\underline{u}$ = m x 1 control vector of player U

$\underline{v}$ = p x 1 control vector of player V

Q_1, Q_2 = m x m and p x p positive definite matrices

Note that C^TC is at least positive semidefinite and that any positive semidefinite matrix may be expressed as the product C^TC.

The state vector $\underline{y}(t)$ is driven by the following dynamic and initial condition

$$\dot{\underline{y}}(t) = F(t)\ \underline{y}(t) + \bar{G}_1(t)\underline{u}(t) + \bar{G}_2(t)\underline{v}(t);\ \underline{y}(0) = \underline{y}_o$$

where $F(t)$ = n x n system dynamic matrix (3.2)

$\bar{G}_1(t)$, $\bar{G}_2(t)$ = n x m and n x p system input distribution matrices

The state vector $\underline{y}$ with dimension n can be reduced to the more convenient and often more meaningful "<u>reduced</u> state vector" $\underline{x}$ with dimension $r \leqslant n$. Define

$$\underline{x}(t) \equiv C\ \Phi(T,t)\ \underline{y}(t) \qquad (3.3)$$

where $\Phi(T,t)$ is the state transition matrix satisfying

$$\dot{\Phi}(T,t) = -\Phi(T,t)\ F(t);\ \Phi(T,T) = I \qquad (3.4)$$

Differentiating (3.3) we get

$$\dot{\underline{x}}(t) = C\dot{\Phi}(T,t)\ \underline{y}(t) + C\ \Phi(T,t)\ \dot{\underline{y}}(t)$$

$$= -C\Phi(T,t)\ F(t)\underline{y}(t) + C\ \Phi(T,t)\ F(t)\ \underline{y}(t)$$

$$+ C\Phi(T,t)\ \bar{G}_1(t)\underline{u}(t) + C\ \Phi(T,t)\ \bar{G}_2(t)\underline{v}(t)$$

$$= G_1(t)\underline{u}(t) + G_2(t)\underline{v}(t) \tag{3.5}$$

where $G_1(t) = C\,\Phi(T,t)\bar{G}_1(t)$

$G_2(t) = C\,\Phi(T,t)\bar{G}_2(t)$

Note that $\underline{x}(0) = C\,\Phi(T,0)\underline{y}(0) \equiv x_o$

$\underline{x}(T) = C\,\Phi(T,T)\underline{y}(T) = C\,\underline{y}(T)$

Thus, the cost function and the dynamic can be rewritten as

$$J(\underline{u},\underline{v}) = 1/2\ \underline{x}^T(T)\underline{x}(T) + 1/2\int_o^T (\underline{u}^T Q_1 \underline{u} - \underline{v}^T Q_2 \underline{v})\,dt \tag{3.6}$$

and

$$\dot{\underline{x}}(t) = G_1(t)\underline{u}(t) + G_2(t)\underline{v}(t);\ \underline{x}(0) = \underline{x}_o \tag{3.7}$$

Since no assumption is made in the derivation of this reduced state form, it is as general as the form represented by equations (3.1) and (3.2). The state $\underline{x}$, however, is a more meaningful measure of the game than $\underline{y}$ since $\underline{x}$ will indicate what the game will end up with if no furthur control is applied by any player.

3.2 Analytical Closed-Loop Solution

Analyses of the problem represented by equations (3.6) and (3.7) will be presented in this section: one with the assumption of existence of a saddle point and another without such assumption.

3.2.1 With Assumption that Saddle Point Exist

The Hamiltonian of the problem is

$$H = 1/2\ (\underline{u}^T Q_1\ \underline{u} - \underline{v}^T Q_2\ \underline{v}) + \underline{\lambda}^T (G_1\ \underline{u} + G_2\ \underline{v}) \tag{3.8}$$

The costate equation is

$$\dot{\underline{\lambda}} = -\frac{\partial H}{\partial \underline{x}} = 0 \Longrightarrow \underline{\lambda} = \text{constant vector}$$

Then

$$\underline{\lambda}(t) = \underline{\lambda}(T) = \underline{x}(T) \tag{3.9}$$

The last equality is obtained from the transversality condition.

Since we are dealing with the unconstrained problem, Pontryagin optimality principle states that

$$\frac{\partial H}{\partial \underline{u}} = Q_1 \underline{u}^* + G_1^T \underline{\lambda} = 0 \tag{3.10}$$

and

$$\frac{\partial H}{\partial \underline{v}} = - Q_2 \underline{v}^* + G_2^T \underline{\lambda} = 0 \tag{3.11}$$

Note that these two equations can be used simultaneously because we have the assumption of existence of a saddle point. Also the positive definitness of Q_1 and Q_2 guarantee that $\underline{u}^*$ and $\underline{v}^*$ is the minimum and the maximum respectively. Thus

$$u^* = - Q_1^{-1} G_1^T \underline{x}(T) \tag{3.12}$$

$$\underline{v}^* = Q_2^{-1} G_2^T \underline{x}(T) \tag{3.13}$$

Substitue $\underline{u}^*$ and $\underline{v}^*$ back into (3.7), integrate from t to T, and solve for $\underline{x}(T)$ we get

$$\underline{x}(T) = W(t)\, \underline{x}(t) \tag{3.14}$$

where

$$W(t) = [I + \int_t^T G_1 Q_1^{-1} G_1^T d\tau - \int_t^T G_2 Q_2^{-1} G_2^T d\tau]^{-1} \tag{3.15}$$

Therefore, the saddle point solution is

$$\underline{u}^*(t) = - Q_1^{-1} G_1^T(t)\, W(t) \underline{x}(t) \tag{3.16a}$$

$$\underline{v}^*(t) = Q_2^{-1} G_2^T(t) W(t) \underline{x}(t) \tag{3.16b}$$

$$J(\underline{u}^*,\underline{v}^*) = 1/2\ \underline{x}^T(t) W(t) \underline{x}(t) \tag{3.16c}$$

3.2.2 Without Assumption that Saddle Point Exist

In this section the minmax and the maxmin solution must be solved separately. With the existence of a saddle point assumption, the condition that the solution exists is that W(t) must be positive definite for all t in the interval [o,T] of the game. We shall see that the condition becomes more stringent without the saddle point assumption.

MINMAX SOLUTION:

First we look for $\underline{v}$ that maximizes $J(\underline{u},\underline{v})$ with an arbitrary $\underline{u}$. For this purpose, equation (3.13) is still valid and we have

$$\hat{\underline{v}}(t) = Q_2^{-1} G_2^T \underline{x}(T) \tag{3.17}$$

Substitue $\hat{\underline{v}}$ into the dynamic equation (3.7), integrate from t to T and solve for $\underline{x}(T)$ we get

$$\underline{x}(T) = P(t)[\underline{x}(t) + \int_t^T G_1(\tau)\ \underline{u}(\tau)d\tau], \quad 0 \leq t \leq T \tag{3.18}$$

where

$$P(t) = [I - \int_t^T G_2(\tau)\ Q_2^{-1}\ G_2^T\ (\tau)\ d\tau]^{-1} \tag{3.19}$$

For P(t) to exist we must have

$$[I- \int_t^T G_2(\tau)\ Q_2^{-1}\ G_2^T\ (\tau)\ d\tau] > 0 \quad \text{for all } 0 \leq t \leq T \tag{3.20}$$

then

$$\underline{\hat{v}}(t) = Q_2^{-1} G_2^T(t) P(t) [\underline{x}(t) + \int_t^T G_1(\tau) \underline{u}(\tau) d\tau] \qquad (3.21)$$

Note that $\underline{\hat{v}}(t)$ is globally optimal as long as (3.20) holds.

Since the starting time t = 0 is arbitrary, rewrite the cost functional in equation (3.6) to complete the game from time t and also substituting (3.19) for $\underline{\hat{v}}$ and manipulate we get

$$J(\underline{u},\underline{\hat{v}}) = 1/2\ \underline{x}^T(T)\ P^{-1}\underline{x}(T) + 1/2 \int_t^T \underline{u}^T Q_1 \underline{u}\ d\tau \qquad (3.22)$$

using (3.18) in (3.22) then

$$J(\underline{u},\underline{\hat{v}}) = 1/2[\underline{x} + \int_t^T G_1 \underline{u} d\tau]^T P[\underline{x} + \int_t^T G_1 \underline{u}\ d\tau] + 1/2 \int_t^T \underline{u}^T Q_1 \underline{u} d\tau \qquad (3.23)$$

Let $\underline{u}(\tau)$ varies by a small amount $\Delta\underline{u}(\tau)$, expand (3.23) to first order in Taylor's series and subtract out nominal terms

$$\Delta J(\underline{u},\underline{\hat{v}}) = \int_t^T \{[\underline{x}(t) + \int_t^T G_1 \underline{u} d\xi]^T P(t) G_1(\tau) + \underline{u}^T(\tau) Q_1\} \Delta\underline{u}(\tau) \qquad (3.24)$$

Since $\Delta\underline{u}(\tau)$ is arbitrary, the variation of the cost functional is zero only when the term {---------} = 0 in equation (3.24) hence

$$\underline{u}(\tau) = -Q_1^{-1} G_1^T(\tau)\ P(t)\ [\underline{x}(t) + \int_t^T G_1 \underline{u} d\xi],\ t \leq \tau \leq T \qquad (3.25)$$

To solve for $\underline{u}$ we multiply each side by $G_1(\tau)$ and integrate from t to T

$$\int_t^T G_1(\tau)\ \underline{u}(\tau)d\tau = -\int_t^T G_1(\tau)Q_1^{-1}G_1^T(\tau)P(t)[\underline{x}(t)+\int_t^T G_1\underline{u}d\xi]d\tau \tag{3.26}$$

Manipulating we get

$$\int_t^T G_1\underline{u}(\tau)d\tau = -[I+\int_t^T G_1Q_1^{-1}G_1^T dr.P(t)]^{-1}.\int_t^T G_1Q_1^{-1}G_1^T d\tau.\ P(t).\underline{x}(t) \tag{3.27}$$

$$\text{Let}\quad A \equiv \int_t^T G_1Q_1^{-1}\ G_1^T\ d\tau\ .P(t) \tag{3.28}$$

Since A is positive semi definite, I + A is positive definite for all $t \in [0,T]$. Therefore the above inverse in (3.27) always exists. Add $\underline{x}(t)$ to both side of (3. 27)

$$\begin{aligned}\underline{x}(t) + \int_t^T G_1\underline{u}(\tau)d\tau &= \underline{x}(t) - [I+A]^{-1}A\ .\ \underline{x}(t)\\ &= \{I-[I+A]^{-1}A\}\ \underline{x}(t)\\ &= [I+A]^{-1}\ \underline{x}(t)\end{aligned} \tag{3.29}$$

Multiplying both side by P(t) we have

$$P(t)[\underline{x}(t) + \int_t^T G_1\underline{u}(\tau)d\tau] + P(t)[I+\int_t^T G_1Q_1^{-1}G_1^T d\tau P(t)]^{-1}x(t) \tag{3.30}$$

$$\text{Let}\quad \bar{A} = \int_t^T G_1Q_1^{-1}G_1^T\ d\tau \tag{3.31}$$

$$\text{and}\quad B = \int_t^T G_2Q_2^{-1}\ G_2^T\ d\tau \tag{3.32}$$

then the right hand side of (3.30) becomes

$$\text{RHS} = [I-B]^{-1}\{I + \bar{A}\,[I-B]^{-1}\}^{-1}\,\underline{x}(t)$$

$$= [I-B+A]^{-1}\,\underline{x}(t)$$

Therefore

$$P(t)\,[\underline{x}(t)+\int_t^T G_1\underline{u}(\tau)d\tau] = [I+\int_t^T G_1Q_1^{-1}G_1^T d\tau-\int_t^T G_2Q_2^{-1}Q_2^T d\tau]^{-1}x(t)$$

$$= W(t)\underline{x}(t) \qquad (3.33)$$

Substitue back into (3.25) yeilds

$$\hat{\underline{u}}(\tau) = -Q_1^{-1}\,G_1^T(\tau)\,W(t)\,\underline{x}(t) \qquad t \leq \tau \leq T \qquad (3.34)$$

Substituting $\hat{\underline{u}}(\tau)$ into (3.20) and perform matrix manipulation similar to the one above

$$\hat{\underline{v}}(t) = Q_2^{-1}\,G_2^T\,(t)\,W(t)\,\underline{x}(t) \qquad (3.35)$$

Comparing (3.17) with (3.35) we have

$$\underline{x}(T) = W(t)\,\underline{x}(t) \qquad (3.36)$$

Using (3.34),(3.35) and (3.36) in (3.22) we get

$$J(\hat{\underline{u}},\hat{\underline{v}}) = 1/2\;\underline{x}^T(t)\;W(t)\underline{x}(t) \qquad (3.37)$$

<u>MAXMIN SOLUTION</u>:

In this case we first look for $\tilde{\underline{u}} = \underset{\underline{u}}{\text{Min}}\;J(\underline{u},\underline{v})$.

Equation (3. 2) can be used to start off

$$\underline{\tilde{u}}(t) = -Q_1^{-1} G_1^T(t)\, \underline{x}(T) \tag{3.38}$$

Substitue $\underline{\tilde{u}}(t)$ into (3.7) integrate and solve for $x(T)$

$$\underline{x}(T) = M(t)\ [\underline{x}(t) + \int_t^T G_2(\tau)\underline{v}\ (\tau)d\tau] \tag{3.39}$$

$$\text{where}\quad M \equiv [I + \int_t^T G_1(\tau)\ Q_1^{-1}G_1^T\ (\tau)]^{-1} \tag{3.40}$$

$$\text{hence}\quad \underline{\tilde{u}}(t) = -Q_1^{-1}G_1^T(t)M\ (t)\ [\underline{x}(t) + \int_t^T G_2(\tau)v(\tau)d\tau] \tag{3.41}$$

Note that M always exists because $\bar{A} \geqslant 0$ and $I+\bar{A} > 0$ for all $0 \leqslant t \leqslant T$. Then

$$\Delta J(\underline{v},\underline{\tilde{u}}) = \int_t^T \left\{[\underline{x}(t) + \int_t^T G_2\underline{v}d\gamma\ ]^T M(t)G_2(\tau) - \underline{v}^T(\tau)Q_2\right\}\cdot \Delta\underline{v}(\tau)d\tau \tag{3.42}$$

for arbitrary $\Delta\underline{v}(\tau)$ again $\{\dots\}$ must be zero for the variation to be zero, hence

$$\underline{\hat{v}}(\tau) = Q_2^{-1}G_2^T(\tau)\ M\ (t)\ [\underline{x}(t) + \int_t^T G_2\ \underline{v}d\gamma\ ],\ t \leqslant \tau \leqslant T \tag{3.43}$$

Premultiply (3.43) by $G_2(\tau)$ and integrate from t to T

$$\int_t^T G_2(\tau)\underline{v}(\tau)d\tau = \int_t^T G_2(\tau)\ Q_2^{-1}G_2^T(\tau)M(t)[\underline{x}(t)+\int_t^T G_2\underline{v}d\gamma\ ]d\tau \tag{3.44}$$

Again (3.43) can be rearranged to give

$$\int_t^T G_2(\tau)\underline{v}(\tau)d\tau = [I - \int_t^T G_2Q_2^{-1}G_2^T\, d\tau\,.M(t)]^{-1}.\int_t^T G_2Q_2^{-1}G_2^T d\tau$$

$$.\; M(t)\underline{x}(t) \qquad (3.45)$$

Now

$$[I - \int_t^T G_2Q_2^{-1}G_2^T d\tau\,.\,M(t)]^{-1} = [(M^{-1}(t) - \int_t^T G_2Q_2^{-1}G_2^T d\tau)M(t)]^{-1}$$

$$= [W^{-1}(t)\; M(t)]^{-1}$$

$$= M^{-1}\;(t)\;W\;(t) \qquad (3.46)$$

Therefore, the indicated inverse in (3.45) exists if W(t) exists which is the same condition for the existence of the saddle point solution. Substitue (3.46) back into (3.45)

$$\int_t^T G_2(\tau)\underline{v}(\tau)d\tau = M^{-1}(t)W(t) \int_t^T G_2(\tau)\; Q_2^{-1}G_2^T(\tau)d\tau M(t)\underline{x}(t) \qquad (3.47)$$

Add $\underline{x}(t)$ to both side, premultiply by $M(t)$, and manipulate to obtain

$$M(t)\;[\underline{x}(t) + \int_t^T G_2(\tau)\underline{v}(\tau)d\tau] = W(t)\underline{x}(t) \qquad (3.48)$$

Hence

$$\underline{\tilde{v}}(\tau) = Q_2^{-1}G_2^T(\tau)\; W(t)\; x(t) \quad \forall \quad t \leqslant \tau \leqslant T \qquad (3.49)$$

Substitue back into (3.41) to get

$$\underline{\tilde{u}}(\tau) = -Q_1^{-1} G_1^T (t)\ W(t)x(t) \tag{3.50}$$

and again $J(\tilde{u},\tilde{v}) = 1/2\ \underline{x}^T(t)\ W(t)\underline{x}(t)$ (3.51)

3.2.3 Summary of Analytical Solutions and Discussion

All the above results may be summarized as follow:

SADDLE POINT SOLUTION

Optimal controls:

$$\underline{u}^*(t) = -Q_1^{-1} G_1^T(t)W(t)\underline{x}(t)$$

$$\underline{v}^*(t) = Q_2^{-1}G_2^T(t)\ W(t)\ \underline{x}(t)$$

Optimal Cost:

$$J(\underline{u}^*,\underline{v}^*) = 1/2\ \underline{x}_o^T\ W(0)\underline{x}_o$$

Condition for existence:

$$I + \int_t^T G_1Q_1^{-1}G_1^T d\tau - \int_t^T G_2Q_2^{-1}G_2^T d\tau > 0 \quad \forall\ 0 \leq t \leq T$$

MINMAX SOLUTION

Optimal Controls:

$$\hat{\underline{u}}(t) = -\ Q_1^{-1}G_1^T(t)W(t)\underline{x}(t)$$

$$\hat{\underline{v}}(t) = Q_2^{-1} G_2^T(t) W(t) \underline{x}(t)$$

Optimal Cost:

$$J(\hat{\underline{u}},\hat{\underline{v}}) = 1/2\ \underline{x}_o^T\ W(0)\underline{x}_o$$

Condition for existence:

$$I - \int_t^T G_2 Q_2^{-1} G_2^T\, d\tau > 0 \quad \forall\ 0 \leq t \leq T$$

MAXMIN SOLUTION

Optimal Controls:

$$\tilde{\underline{u}}(t) = -Q_1^{-1} G_1^T(t) W(t) \underline{x}(t)$$

$$\tilde{\underline{v}}(t) = Q_2^{-1} G_2^T(t) W(t) \underline{x}(t)$$

Optimal Cost:

$$J(\tilde{\underline{u}},\tilde{\underline{v}}) = 1/2\ \underline{x}_o^T\ W(0)\underline{x}_o$$

Condition for existence:

$$I + \int_t^T G_1 Q_1^{-1} G_1^T d\tau - \int_t^T G_2 Q_2^{-1} G_2^T d\tau > 0 \ \forall 0 \leq t \leq T$$

DISCUSSION:

Following remarks may be made about these solutions:

(1) Controls and costs are the same in each case, thus a saddle point does indeed exist for linear-quadratic differential game. The optimal cost in each case is the value of the game and neither player can do anything unilaterally to improve his cost.

(2) Conditions for the existence of the sadd. . point solution and the maxmin solution are the same. For minmax solution, however, the condition is more stringent. It might be noted that if the necessary and sufficient for the minmax solution is satified for a game, then the necessary and sufficient condition for both the maxmin and the saddle point solution will be automatically satisfied because the missing matrix in the conditions for the latter two is at least positive semidefinite.

(3) All the solutions solved for this problem are in closed-loop form which is indeed more desirable than the open-loop solution. It might be noted, however, that linear-quadratic problem is about the only type of differential game for which closed-loop solution may be solved analytically. Moreover, as we shall see later, if the constraints on state and controls are added to the game, even for a linear-quadratic problem a closed-loop solution is not always guaranteed.

3.3 An Illustrative Example Without Control Constraint

The example given in this section will be the same as the one considered by McFarland. However, more extensive results will be offered to gain a more meaningful insight into the problem. Our goal here is to show that in contrary to McFarland's implication that the linear-quadratic formulation of a pursuit-evasion problem is likely to yield trivial solution, this difficulty may be avoided by careful examination of the problem to avoid any conjugate point in the solution. The incentive to present the illustrative example in this manner is two folds. First it will be shown that even simple unconstrained linear-quadratic problem can have a meaningful physical realization. Also in the next section, the same example with constraints will be used to show that analytical solutions are not possible in such case and numerical solutions with and without the saddle point assumption will be offered.

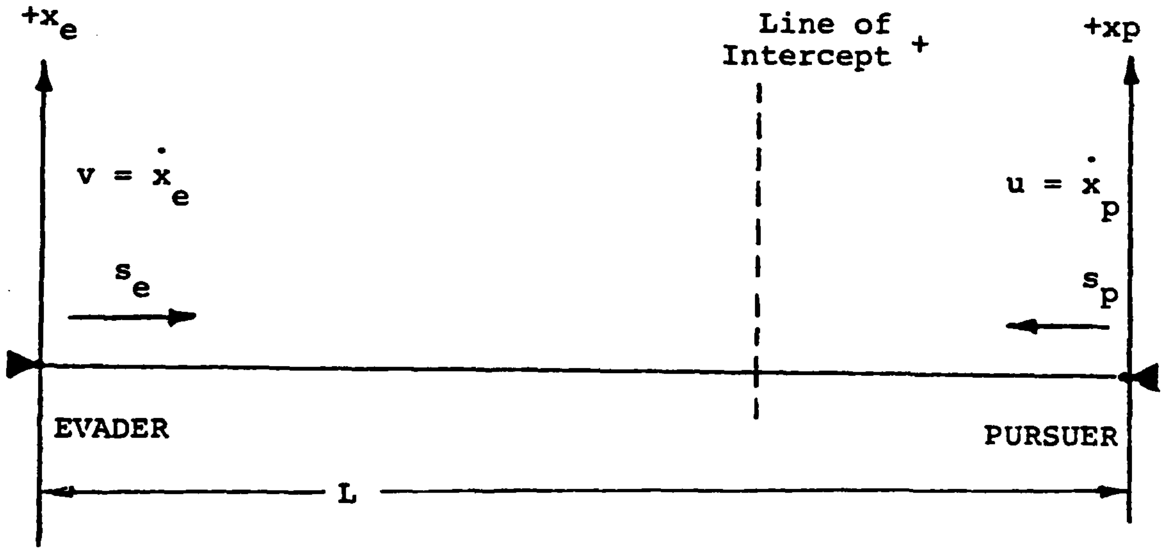


Figure 5. Simple Plannar Intercept Problem

A simple planar intercept problem is diagrammed in figure 5. x_e and x_p are lateral positions of each players who move towards one another with constant forward velocities s_e and s_p respectively. The interception time or the time when the players pass each other is $T = \frac{L}{S_e + S_p}$ seconds. Each player controls his lateral position by using his respective lateral velocities, u(t) and v(t), as control inputs. Thus we have

$$\begin{aligned} \dot{x}_p &= u, \quad x_p(0) = 0 \\ \dot{x}_e &= v, \quad x_e(0) = 0 \end{aligned} \tag{3.52}$$

Generally, x_p and x_e can be used as the states of the problem. But as we shall see, it is more convenient and more meaningful to define a state x(t) as

$$x(t) = x_e(t) - x_p(t) \tag{3.53}$$

Thus x(t) may be interpreted as the <u>lateral miss distance</u> if both players use no further control from time t until the end of the game. Using (3.53) in (3.52) the dynamic equation is

$$\dot{x}(t) = v(t) - u(t), \; x(0) = 0 \tag{3.54}$$

The pursuer is trying to minimize the miss lateral distance at the time of interecption without using excessive control energy while the evader is trying to maximize the same miss lateral distance while using reasonable control. Therefore, we have a two person zero-sum differential game with the cost function:

$$J(u,v) = 1/2\, x^2(T) + 1/2 \int_t^T (q_1 u^2 - q_2 v^2)\,dt \qquad (3.55)$$

Using the results of section (3.2), the solutions are

$$\hat{u}(t) = \tilde{u}(t) = u^*(t) = \frac{W(t)}{q_1}\, x(t) \qquad (3.56a)$$

$$\hat{v}(t) = \tilde{v}(t) = v^*(t) = \frac{W(t)}{q_2}\, x(t) \qquad (3.56b)$$

where

$$W(t) = \left(1 + \frac{T-t}{q_1} - \frac{T-t}{q_2}\right)^{-1} \qquad (3.57)$$

The necessary and sufficient condition for the minmax solution is

$$1 - \frac{T-t}{q_2} > 0 \quad \text{for} \quad 0 \leqslant t \leqslant T \qquad (3.58)$$

This is the same as the condition $q_2 > T$. Substituting (3.56) into (3.54) it is apparent that the only stable solution for the resulting differential equation is $x(t) = 0$ for $0 \leqslant t \leqslant T$. Therefore

$$\hat{u}(t) = \tilde{u}(t) = u^*(t) = 0 \qquad (3.59a)$$

$$\hat{v}(t) = \tilde{v}(t) = v^*(t) = 0 \qquad (3.59b)$$

This solution makes sense in the pursuer viewpoint since the initial lateral miss distance is zero and since the evader is not making any move, the pursuer then can hold his position until he runs into the evader at the time of interception. From the evader's point of view, however, this is indeed a strange solution since we would expect him to do something to avoid collision with the pursuer.

This strange result occurs because (3.58) calls for too much weight q_2 on control v(t) otherwise we would have the maxmin solution $\tilde{v}(t) \longrightarrow \infty$. However, we only have this dilemma for $x(0) = 0$ which is the only case McFarland considered. If we let $x(0) = x_o \neq 0$, then the solutions are

$$\hat{u}(t) = \tilde{u}(t) = u^*(t) = \frac{q_2}{q_1 q_2 - (q_1 - q_2)T} x_o \quad (3.60a)$$

$$\hat{v}(t) = \tilde{v}(t) = v^*(t) = \frac{q_1}{q_1 q_2 - (q_1 - q_2)T} x_o \quad (3.60b)$$

with the value of the game

$$J(\hat{u}, \tilde{v}) = \frac{q_1 q_2}{2(q_1 q_2 - (q_1 - q_2)T} x_o^2 \quad (3.61)$$

The interpretation of this result can be summarized as follow:

Case 1: $q_1 = q_2 = q$

then $\hat{u} = \tilde{u} = u^* = \frac{1}{q} x_o$

$\hat{v} = \tilde{v} = v^* = \frac{1}{q} x_o$

and $J(\hat{u}, \tilde{v}) = 1/2 \; x_o^2$

In this case the evader cannot get further away from the initial lateral displacement if the evader is using his optimal control.

Case 2: $q_1 > q_2$

then from (3.60) $|\hat{u}(t)| < |\tilde{v}(t)|$. Physically, this makes sense because in this case since the pursuer is putting more weight on his control, he is penalized more than the evader if both players use the same amount of control. Therefore, the pursuer is induced to use less control than the evader. Also since

$$x(T) = \frac{q_1 q_2}{q_1 q_2 - (q_1 - q_2)T} x_o \qquad (3.62)$$

$x(T)$ is larger than x_o in this case. Moreover, the larger q_1 is in relative to q_2, the larger $x(T)$ will be. Thus the pursuer can escape if q_1 is large enough and the evader is restricted to use a small amount of control. It is interesting to note that, if the necessary and sufficient condition in equation (3.58) is satisfied, $x(T)$ cannot be negative with respect to x_o.

Case 3: $q_1 < q_2$

then $$|\hat{u}(t)| > |\tilde{v}(t)|$$

the evader is induced to use less quantity of control in this case because more weight is being put on his control. From (3.62), $x(T)$ is smaller than x_o in this case and the pursuer can get closer to the evader than the initial lateral displacement. Interception can be made if q_2 is large enough.

The magnitude of q_2 required for interception depends upon the radius of interception, the magnitude of x_o, the time of interception T, and the weight q_1 on the pursuer control.

This example clearly illustrates that even a simple unconstrained linear-quadratic problem can be meaningful if it is set up carefully to avoid the conjugate point difficulty.

3.4 Linear Quadratic Problem with Hard Limit on Controls

In this section, we shall use the same illustrative problem described in section 3.3. The cost function and the dynamic equation are repeated here for convenience.

$$\text{Cost:} \quad J(u,v) = 1/2\, x^2(T) + 1/2 \int_t^T (q_1 u^2 - q_2 v^2)\,dt \qquad (3.63)$$

$$\text{Dynamic:} \quad \dot{x} = v(t) - u(t),\; x(0) = x_o \qquad (3.64)$$

In addition we add the following constraints

$$|u| \leqslant 1 \quad , \quad |v| \leqslant 1 \qquad (3.65)$$

This problem can be solved analytically, if the parameters are set up in such a way that u and v do not exceed their limits. In that case the results are of course the same as those presented in the last section. The actual derivation for the conditions and the solutions for this problem such that the optimal controls lie within the control boundaries will be taken up in appendix B. Also in appendix B, we shall demonstrate the equivalency between the closed loop and the open loop solutions for this specific problem.

However, this problem in general cannot be solved analytically. To illustrate this point, let us try to find the minmax solution. The Hamiltonian of the problem is

$$H = 1/2(q_1u^2 - q_2v^2) + \lambda(v-u) \qquad (3.66)$$

and the costate equation is

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = 0 \qquad (3.67)$$

Therefore, $\lambda(t) = \text{constant}$ (3.68)

From transversality condition: $\lambda(T) = x(T)$ (3.69)

Thus $\lambda(t) = \lambda(T) = x(T) \;\forall\; 0 \leqslant t \leqslant T$ (3.70)

Now, for an arbitrary u, maximize J with respect to v.

Pontryagin Maximal Principle states that

$$H(x^*, u, v^*, \dot{\lambda}^*, t) \geqslant H(x,u,v,\lambda^*,t) \qquad (3.71)$$

where * indicates optimal quantities. Consider the terms in the Hamiltonian which contain v, we have

$$-1/2\; q_2v^2 + \lambda v \qquad (3.72)$$

if $q_2 < \lambda$, it can be shown that

$$v^* = \text{sgn}\,\lambda = \text{sgn}\; x(T) \qquad (3.73)$$

substituting (3.73) back into (3.64) yields

$$\dot{x} = \text{sgn}\; x(T) - u \qquad (3.74)$$

integrate from 0 to T and rearrange

$$x(T) = x_o + T\,.\,\text{sgn}\; x(T) - \int_t^T u\,dt \qquad (3.75)$$

It is not clear that x(T) and or sgn x(T) can be solved from (3.75) unless all parameters including the arbitrary u are assigned numerical values. In fact equation (3.75) is transcendental.

This problem then must be solved by numerical methods. In the next section, two numerical solutions will be offered, one with the assumption of a saddle point DDP is used directly to simultaneously solve for u^* and v^*, and the other without the assumption of a saddle point the algorithm developed in the last chapter is used to solve for the maxmin and the minmax solutions.

3.5 Numerical Solutions

Computer programs using Fortran Language in conjunction with the WATFIV compiler are written to obtain numerical solutions for the problem described in the last section. These programs are listed in Appendix C. In the first program, the existence of a saddle point is assumed, and DDP is used to simultaneously solve for the optimal controls for each player. In the other two programs, the minmax and the maxmin solutions are searched for using the algorithm developed in Chapter 2. As expected the program with the saddle point assumption contains less number of programming steps than each of the other two programs. We shall call the solution obtained with saddle point existence assumption the saddle point solution, and the other two the minmax solution and the maxmin solution respectively for obvious reason. A large number of batch jobs are computed using UCLA Campus Computing Network's IBM System 360 Model 91. The computation time for all three programs are extremely fast. The execution time for all three types of solutions are essentially the same. For a typical set of parameters, the execution time for all three programs are approximately 0.2 second each for an 8 seconds encountered between the two players.

3.5.1 Algorithm Mechanization

An integration scheme is needed to mechanized the algorithm for the DDP both in integrating the state equation forward and also to integrate the set of equations (2.49) backward. Since the structure of this problem calls for a constant values for the optimal controls during the entire interval of the game, simple Euler's scheme of integration can be used to obtain accurate results.

To mechanize the algorithm on the computer, discretization must be made. For this purpose, the encountered time is devided into 64 increments. For a typical encountered time of 8 seconds then each increment of time is equivalent to one-eighth of a second.

Even though the programs are written to accommodate the step-size adjustment described in section 2.5.3, no step-size adjustment were needed for the large set of parameters on the trial runs on this problem. Equation (2.52) is satisfied in all cases of the trial runs.

Table 1,2, and 3 are computer printouts of the saddle point solution, the minmax solution, and the maxmin solution respectively for the following set of parameters:

$$L = 180 \text{ kilofeet}$$

$$S_e = 15 \text{ kilofeet per second}$$

$$S_p = 7.5 \text{ kilofeet per second}$$

$$T = \frac{L}{S_e+S_p} = 8 \text{ seconds}$$

The control limits are chosen as ten percent of their respective forward velocities

$$|v| \leq 1.5 \text{ kilofeet per second}$$

$$|u| \leq .75 \text{ kilofeet per second}$$

For the saddle point solution, both initial controls were chosen as zero. USTAR and VSTAR are the controls that minimizes and maximizes respectively the Hamiltonian in each iteration. Number "1" in the "step adj" column indicates that the set of equations (2.49) is integrated backward to the time $t = 0$. It might be noted here that the further the algorithm progresses, the closer the predicted cost change in the column "A(N)" agrees with the actual cost change in the column "DELJ". For this set of parameters, the saddle point solution converges in five iterations with approximately 0.2 second execution time. Both optimal controls are saturated for this set of parameter. The value VSTAR = 1 in the first iteration satisfies equation (3.60 b) of the unconstrained problem. Therefore, similar to optimal control, the results here confirmed that a constrained differential game cannot be solved as an unconstrained differential game

~~saddle point solution~~

XO=10.0 Q1=10 Q2=10 MAXU=0.75 MAXV=1.50 JO= 50.00

ITERATION	USTAR	VSTAR	FINALX	STEP ADJ	A(N)	DELJ	COST
1	0.75	1.00	12.00	1	2.500	4.500	54.50
2	0.75	1.20	13.60	1	1.600	2.880	57.38
3	0.75	1.36	14.88	1	1.024	1.843	59.22
4	0.75	1.49	15.90	1	0.655	1.180	60.40
5	0.75	1.50	16.00	1	0.093	0.097	60.50

ALGORITHM CONVERGES N= 1
USTAR=0.75KFT/SEC VSTAR=1.50KFT/SEC FINALX= 16.00 COST=60.50

Table 1. Typical run of a saddle point solution for the specified set of parameters.

MINMAX SOLUTION

XO=10.0 Q1=10 Q2=10 MAXU=0.75 MAXV=1.50 JO= 54.50

ITERATION	USTAR	VSTAR	FINALX	STEP ADJ	A(N)	DELJ	COST
1	0.75	1.20	13.60	1	1.600	2.880	57.38
2	0.75	1.36	14.88	1	1.024	1.843	59.22
3	0.75	1.49	15.90	1	0.655	1.180	60.40
4	0.75	1.50	16.00	1	0.093	0.097	60.50

2 UO= 0.75 VO= 1.50 COST= 60.50 MAX COST GRADIENT= 8.50

1	0.75	1.50	16.00	1	0.000	0.000	60.50

ALGORITHM CONVERGES N= 1
USTAR=0.75KFT/SEC VSTAR=1.50KFT/SEC FINALX= 16.00 COST= 60.50

Table 2. Typical run of a minmax solution for the specificed set of parameters.

```
MAXMIN SOLUTION

XO=10.0   Q1=10   Q2=10   MAXU=0.75   MAXV=1.50   JO= 54.50

ITERATIONUSTAR      VSTAR      FINALX     STEP ADJ  A(N)       DELJ       COST
 1        0.75       1.00       12.00       1            0.000      0.000      54.50

 2  UO= 0.75  VO= 1.20  COST= 57.38  MINCOST GRADIENT=    2.00
 1        0.75       1.20       13.60       1            0.000      0.000      57.38

 3  UO= 0.75  VO= 1.36  COST= 59.22  MINCOST GRADIENT=    1.60
 1        0.75       1.36       14.83       1            0.000      0.000      59.22

 4  UO= 0.75  VO= 1.49  COST= 60.40  MINCOST GRADIENT=    1.28
 1        0.75       1.49       15.90       1            0.000      0.000      60.40

 5  UO= 0.75  VO= 1.50  COST= 60.50  MINCOST GRADIENT=    1.02
 1        0.75       1.50       16.00       1            0.000      0.000      60.50

ALGORITHM CONVERGES  N= 1
USTAR=0.75KFT/SEC  VSTAR=1.50KFT/SEC  FINALX= 16.00  COST= 60.50
```

__Table 3.__ Typical run of a maxmin solution for the specified set of parameters.

and let the controls saturated when and if the resulting controls exceed their limits. These values, however, can be used as the initial controls for the algorithm as illustrated in the minmax and the maxmin solutions.

In both the minmax and the maxmin solutions, the initial controls are computed from equations (2.60a) and (2.60b), the saturated value is used whenever a control exceeds its limit. In the minmax solution, the gradient of the maximum cost in each of the overall iteration is always negative. Similarly, in the maxmin solution, the gradient of the minimum cost is always positive. This indicates that the right directions are being searched. Note also that the absolute values of the gradients form monotonic decreasing sequences and thus assure the convergence property of the algorithm. In the minmax and the maxmin algorithm, DDP is used for the inner optimization, and gradient projection method is used in the outer or overall optimization. For this particular set of parameters, the minmax solution converges in two overall iterations with four iterations of the DDP for the first inner maximization while the maxmin solution requires five overall iterations to converge but each inner minimization converges in one iteration of the DDP. The total computation time for both solutions are again approximately 0.2 second each. Therefore, we can conclude that there is no appreciable difference in the computation time of this problem for all three types of solutions.

For all the large number of sets of parameters run for this problem, all three solutions give the same answers for the optimal controls. Therefore, even though it has not been vigorously proved analytically, we may heuristically say that the saddle point does indeed exist for this type of constrained linear-quadratic differential game as confirmed by our numerical experiments.

For the particular set of parameters shown on table 1 through 3, equal weights are put on the penalties on the controls of both player. In this case, we recall that the unconstrained case calls for an equal amount of controls from both players and neither the pursuer can get any closer nor the evader can manuver to be further away than the initial lateral displacement x_o. In the constrained case, however, since the pursuer in this case is more limited in his lateral speed, the evader can use his superior capability to get further away than the initial lateral displacement as shown by $x(T) = 16$ kilofeet when $x_o = 10$ kilofeet in this case.

3.5.2 Effects of Parameter Variations

Table 4 illustrates the effects of changing the initial condition x_o with a fixed set of other parameters. As expected, when the initial lateral displacement is small, the solutions stay within the boundaries and are the same as those obtained in the unconstrained case. For the set of parameters shown in table 4, the solutions are the same for both the constrained and the inconstrained case for $|x_o| \leqslant .75$ kilofeet.

With larger initial lateral displacement, the pursuer's control becomes saturated. The evader can then take advantage of his superior capability to obtain larger final lateral separation between the two; whereas we have noted before in the unconstrained case that for $q_1 = q_2$, which is the case here, neither player can get any closer nor further away from each other than their initial lateral displacement. Besides making use of his superior capability, the evader has another reason which induces him to use more control in this case than he would have used in the unconstrained case. That reason is the fact that the control limited by the pursuer has introduced a relative saving in the cost function for the evader.

The evader's control becomes saturated when x_o is only 9 kilofeet whereas in the unconstrained case this same x_o would yield an optimal control of only .9 kilofeet per second for the evader which is only sixty per cent of his capability. For $|x_o| \geqslant 9$ kilofeet both players use their maximum capabilities for their optimal controls. The lateral missdistance is 6 kilofeet greater at the final time than it was at the initial time. This difference is brought about by the evader's superior capability and remains the same for all $|x_o| \geqslant 9$ kilofeet.

Table 5 and 6 show the effects of changing the pursuer's weighting factor q_1 when the evader's weighting factor $q_2 = 10$ and $x_o = 8$ kilofeet and 10 kilofeet respectively. The control limits in both cases are $|u| \leqslant .75$ kilofeet per second and $|v| \leqslant 1.5$ kilofeet per second. In table 5 we see that the pursuer's control is saturated for $q_1 \leqslant 12$. This is not surprising because with relatively small q_1, the gain in the final lateral miss distance offset the penalty of using more control for the pursuer thus he would use as much control as he possibly could. With larger q_1, however the pursuer is forced to use less control than his limit. The solutions for $q_1 > 12$ in table 5 are the same as the unconstrained case and the minmax and maxmin solutions converge in one iteration. In table 6 both players are forced to use their respective maximum control because of the relatively large value of x_o.

In table 7, the values of x_o, q_1, and q_2 are doubled when compared to the same parameters in table 5. Close examimation reveals that the solutions in both table 5 and table 7 follows the same relative pattern even though the absolute magnitude of the unsaturated controls for both players are lower in table 7 because of greater penalties for the control inputs.

Table 8 and 9 demonstrate the effects of changing the control limits. In table 8, both players have equal capabilities, the optimal controls in this case then depend upon the relative values of the penalty weights q_1 and q_2

and the initial lateral displacement x_o. In table 9, the pursuer's lateral capability exceeds that of the evader. The limits on control inputs are interchanged if compared to those in table 7, the pattern of the solutions, however, is consistent if the limits interchange is taken into account.

3.5.3 Discussion on the Algorithms

Before we close this chapter, several points can be made on the algorithms used in this section.

(1) All three types of solution are the same for each particular set of parameters. Therefore, we can conclude that for linear-quadratic problem saddle point exists for both the constrained and the unconstrained cases.

(2) The saddle point solution takes less programming steps than each of the minmax solution and the maxmin solution.

(3) Computation times are approximately the same for all types of solution. All three types converge very rapidly in most cases.

(4) The saddle point solution uses $u(t) = 0$ and $v(t) = 0$ as initial controls whereas the minmax and the maxmin solutions use the results of the unconstrained case as initial controls (using saturated values wherever appropriated). This, however, is a very minor modification since the solutions for the unconstrained case is very easy to compute.

For $q_1 = q_2 = 10$ u .75 v 1.5

x_o	u^*	v^*	$x^*(T)$	J^*
1.0	.10	0.10	1.00	0.50
2.5	.25	0.25	2.50	3.13
5.0	.50	0.50	5.00	12.50
7.5	.75	0.75	7.50	28.13
8.0	.75	1.00	10.00	32.50
9.0	.75	1.50	15.00	45.00
10.0	.75	1.50	16.00	60.50
11.0	.75	1.50	17.00	77.00
12.0	.75	1.50	18.00	94.50
15.0	.75	1.50	21.00	153.00
20.0	.75	1.50	26.00	270.50

Table 4. Effects of Variation in x_o

x_o = 8 kft q_2 = 10   u .75   v 1.5

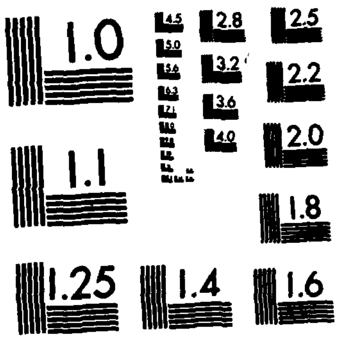
q_1	u^*	v^*	$x^*(T)$	J^*
1	.75	.98	9.86	12.25
2	.75	.98	9.86	14.50
4	.75	.98	9.86	19.00
6	.75	.98	9.86	23.50
8	.75	.99	9.89	28.00
10	.75	.99	9.89	32.50
12	.75	.99	9.90	37.00
14	.74	1.04	10.37	41.48
16	.71	1.14	11.43	45.71
18	.69	1.24	12.41	49.65
20	.67	1.33	13.33	53.33

Table 5. Effects of variation in q_1 with x_o = 8 kft

AD-A124 643 ALGORITHMS FOR DIFFERENTIAL GAMES WITH BOUNDED CONTROL AND STATES(U) CALIFORNIA UNIV LOS ANGELES SCHOOL OF ENGINEERING AND APPLIED SCIENCE A CHOMPAISAL MAR 82 2/2

UNCLASSIFIED DASG60-80-C-0007 F/G 12/1 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

x_o = 10 kft. q_2 = 10 u .75 v 1.5

q_1	u^*	v^*	$x^*(T)$	J^*
1	.75	1.5	16.0	40.25
2	.75	1.5	16.0	42.50
4	.75	1.5	16.0	47.00
6	.75	1.5	16.0	51.50
8	.75	1.5	16.0	56.00
10	.75	1.5	16.0	60.50
12	.75	1.5	16.0	65.00
14	.75	1.5	16.0	69.50
16	.75	1.5	16.0	74.00
18	.75	1.5	16.0	78.50
20	.75	1.5	16.0	83.00

Table 6. Effects of variation in q_1 with x_o = 10 kft.

x_o = 16 q_2 = 20 u .75 v 1.5

q_1	u^*	v^*	$x^*(T)$	J^*
1	.75	0.83	16.58	85.57
5	.75	0.83	16.58	94.57
10	.75	0.83	16.58	105.82
15	.75	0.83	16.58	117.07
20	.75	0.83	16.58	128.32
25	.70	0.87	17.38	139.12
30	.62	0.92	18.46	147.68
35	.55	0.96	19.31	154.47
40	.50	1.00	19.97	159.99

<u>Table 7</u>. Solutions when x_o, q_1, and q_2 are doubled as compared to those values in Table 5.

x_o = 16 q_2 = 20 u 1.0 v 1.0

q_1	u^*	v^*	$x^*(T)$	J^*
1	1.00	0.67	13.34	57.33
5	1.00	0.67	13.34	73.33
10	1.00	0.67	13.34	93.33
15	0.94	0.71	14.12	112.94
20	0.80	0.80	16.0	128.00
25	0.70	0.87	17.38	139.12
30	0.62	0.92	18.46	147.68
35	0.55	0.96	19.31	154.47
40	0.50	1.00	19.97	159.99

Table 8. Solutions when both players have equal capabilities.

$x_o = 16 \quad q_2 = 20 \quad u \quad 1.5 \quad v \quad .75$

q_1	u^*	v^*	$x^*(T)$	J^*
1	1.50	0.34	6.68	22.33
5	1.46	0.36	7.26	58.18
10	1.14	0.57	11.42	91.43
15	0.94	0.71	14.12	112.94
20	0.79	0.75	15.71	127.86
25	0.66	0.75	16.69	138.33
30	0.58	0.75	17.37	146.04
35	0.51	0.75	17.90	151.97
40	0.46	0.75	18.33	156.66

Table 9. Solutions when the pursuer's capability exceeds that of the evader

CHAPTER 4

A NONLINEAR STOCHASTIC PURSUIT - EVASION PROBLEM

The most natural application of differential game theory probably falls on a class of problem known as pursuit-evasion where two or more adversaries engage in a combat type mission. The state of the art of this problem has already been discussed in Chapter 2 of this report.

In this chapter, a model for nonlinear stochastic pursuit-evasion two-person zero-sum differential game will be formulated. The problem will then be solved using the simple algorithm developed in Chapter 2 for a set of designated parameters. Lastly, many aspects of the computational results will be compared to those obtained by McFarland.

Several important features of two person zero-sum differential games will be illustrated by the problem studied in this chapter. The dynamics of the problem are nonlinear using the set of sufficient statistic of the actual physical entities. In this manner the elements in the set of sufficient statistics can be treated as the state variables of a deterministic problem and hence reduce the complexities of the stochastic problem greatly. Moreover, the values of the cost function for the minmax and the maxmin solutions of this problem are not the same. Thus, we are presented with a realistic problem whose solutions are not "saddlepoint" and hence substantiating the fact that saddlepoint does not have to exist in a general differential game. This fact also serves to strengthen the two examples presented in Chapter 2. The cost function of the problem utilizes the probability of survival as a probabilistic measure and provides a realistic flavor of a stochastic differential game. Furthurmore, the information sets available to each player are limited on only those state variables observable by each player.

4.1. Description of the Problem

A simplification of the missle-anti-missle intercept problem will be studied in this chapter. An incoming attacking missle, maneuverable laterally, is trying to avoid being intercepted by an antimissle, also maneuverable laterally. The attacking missle, however, is also charged with the task of trying to destroy an isolated target (a military installation, an industrial complex, or any other strategic target) and thus cannot stray too far away from a designated path. On the other hand, the antimissle which is trying to defend the target is launched from an area on or near the target. A ground support radar will keep track of the position of the oncoming attacking missle and hence the defender will have a full set of informations on both his own and the enemy positions. The target defender will make use of these informations and try to minimize the distance of closest approach between it and the intruder. If it gets close enough, the attacking missle is neutralized or captured

Only one pass is allowed for this problem because once the missles pass one another, the antimissle will not be able to turn around and try to catch the attacking missle.

The control center of the attacking missle will be too far away to observe the actual positions of both missles by radar. However, with the present technology, it is not hard to visualize an attacking missle with an on board computing capability to compute its own displacement from a designated path. Therefore, the attacking missle will only be able to make use of the information on his own position. The attacking missle is deemed to score or accomplish its mission if it manages to avoid interception and yet reach the target zone.

For this problem, we shall call the evader player U and the interceptor player V.

To make the problem tractable, simplification assumptions will be made, nevertheless significant features of the general problem will be maintained. The simplified version of the intercept problem is illustrated in Figure 6. One simplified assumption is that planar motion is assumed. This is equivalent to a classical aeriel combat encounter over a flat earth.

The mean initial line of sight (LOS) between the two players has the length L. This line will be used as a basic reference line for the problem. The initial position

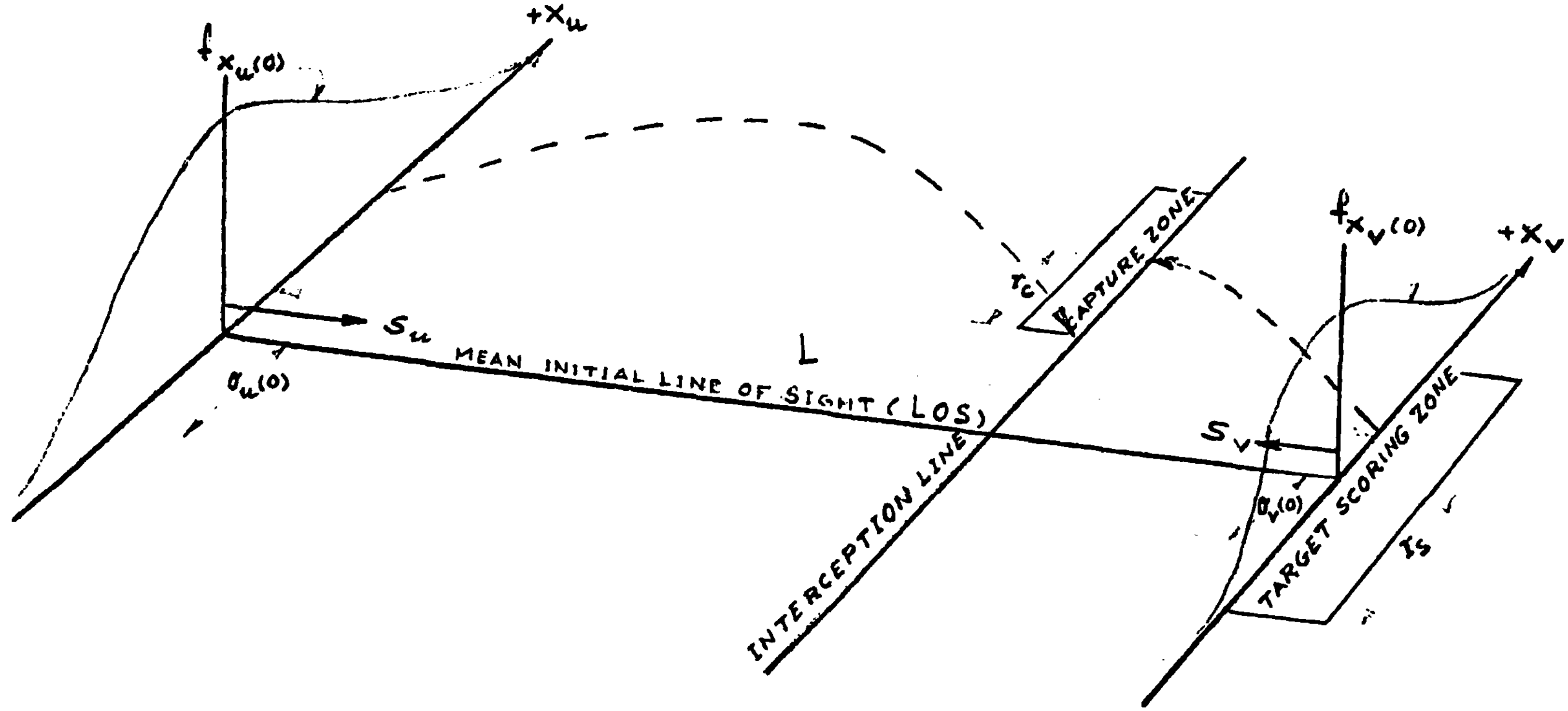


Figure 6. Schematic of Stochastic Pursuit-Evasion Problem

dispersion along the reference line L is much less important to the problem than the lateral position dispersion, since the variation in L only effects the variation in the "interception time T" and the "target engagement time T_S". The interception time T is defined as the time when both players reach the locus of distance of closest approach. The target engagement time T_S is defined as the time when player U reach the line extended from the target perpendicular to the line of sight L. Thus T and T_S may be regarded as fixed.

Both players' initial velocities are assumed parallel to the line L. The lateral maneuvering of each player is assumed uncoupled to the forward motion. This assumption is not a serious determent to the realism of the intercept problem since the lateral displacement will typically not exceed 5% of L on either side of the mean initial LOS.

The initial lateral position of each player is assumed a random variable normally distributed about L with the mean equal to zero. The uncertainty in the lateral position of the attacking missle arises from accumulated error picked up during the launch and midcourse preengagement phases of ICBM flight whereas the uncertainty in the lateral position of the defending antimissle is dued to the inaccuracy in controlling the violent acceleration subjected during the launch phase which is assumed prior to commencement of this problem.

The capture zone r_s is a measure of potency of the interceptor. The width of this zone depends upon the characteristic of the proximity fuse used in the warhead of the interceptor. The scoring zone r_s is a measure of both the potentcy of the attacking ICBM and the vulnerability of the target. The width of r_s depends upon the explosive characteristic of the warhead of the ICBM and the vulnerability of the target being attacked. The target is considered destroyed if the ICBM can get within the scoring zone at the time T_S. Normally r_c will be much smaller than r_s.

4.2 Formulation of the Problem

The missle antimissle problem described in the above section will be formulated as a nonlinear stochastic differential game as follows:

4.2.1 Dynamics of the Problem

The parameters that are important to both players are their respective lateral positions normal to the line L. As mentioned before, the lateral maneuvering is assumed uncoupled to the forward motion, and the players are assumed able to maneuver laterally by controlling their lateral velocities:

$$\dot{x}_u(t) = c_u(t); \quad x_u(o) = x_{uo} \qquad \text{..............(4.1a)}$$

$$\dot{x}_v(t) = c_v(t); \quad x_v(o) = x_{vo} \qquad \text{..............(4.1b)}$$

where $c_u(t)$ and $c_v(t)$ are instataneous lateral velocities controlled by U and V respectively. The initial conditions x_{uo} and x_{vo} are random variables are normally distributed with zero mean and the covariances σ_u^2 (0) and σ_v^2 (0). These probability density functions are shown by equation (4.2).

$$f(x_{uo}) = \frac{1}{\sigma_u(0)\sqrt{2\pi}} \exp\left[-x_{uo}^2 / 2\,\sigma_u^2(0)\right] \ldots\ldots (4.2a)$$

$$f(x_{vo}) = \frac{1}{\sigma_v(0)\sqrt{2\pi}} \exp\left[-x_{vo}^2 / 2\,\sigma_v^2(0)\right] \ldots\ldots (4.2b)$$

The lateral velocities $c_u(t)$ and $c_v(t)$ are then functions of random processes $x_u(t)$ and $x_v(t)$. These velocities are limited to within 10% of their associated average forward speeds to validate the uncoupled assumption.

The vector $\underline{x}^T(t) = [x_u(t) \mid x_v(t)]$ is assumed to be a Gauss Markov process where only two statistics, a mean and a convariance, are needed to specify it completely. To make this assumption valid, the system that generates the process must be linear. Thus we are required to choose $c_u(t)$ and $c_v(t)$ as linear functions of $x_u(t)$ and $x_v(t)$.

For the interceptor, player V, the important quantity that will have to be minimized is the lateral distance between him and the attacking missle U at the interception time $x_u(T) - x_v(T)$. However, at any particular time before the interception time $t < T$, $x_u(T) - x_v(T)$ is not available to V. Therefore, V has no choice but to use the most recent corresponding information that is the best indicator for $x_u(T) - x_v(T)$, namely $x_u(t) - x_v(t)$. Hence $c_v(t)$ is defined as

$$c_v(t) = v(t)[x_u(t) - x_v(t)] \quad \ldots\ldots (4.3)$$

where $v(t)$ = feedback gain function, to be found as V's control

$x_u(t), x_v(t)$ = current states of each players

For the attacking missle, player U, the most important measure for him is the distance by which he misses the target, $x_u(T_s)$, at the time when he crosses the target boundary. Between the interception time T and the target engagement time T_s, U is not at all effected by any action on V's part during this interval. Therefore, the problem in this interval is an optimal control problem with only one player U starting from an initial state $x_u(T)$ and minimizing the final state $x_u(T_s)$ using "reasonable" control along the way. The problem in this duration can then be solved as a linear quadratic problem with the result

$$x_u(T_s) = k\, x_u(T), \quad o < k < 1 \quad \ldots\ldots. (4.4)$$

where the fraction k depends upon the time duration $T_s - T$ and the weight on the control u(t). For this differential game then, we shall assume that U can reduce $x_u(T)$ by a given fraction k during the interval $[T,T_s]$. Ideally U would like to have his feedback function as a function of $x_u(T_s)$ (since he cannot observe the state $x_v(t)$ at any time).

Hence

$$C_u(t) = u(t) \;\; x_u(T_s) \qquad \ldots\ldots\ldots \; (4.5)$$

with

u(t) = feedback gain function, to be found as U's control

using (4.4) in (4.5)

$$C_u(t) = u(t)\; k\; x_u(T) \qquad \ldots\ldots\ldots\ldots (4.6)$$

again it is obvious that U do not have $x_u(T)$ at any time prior to the time of interception equation (4.6) is not causal. Therefore, the best he could do is to use the most current information $x_u(t)$ instead of it. Hence we have

$$C_u(t) = u(t)\; k\; x_u(t) \qquad \ldots\ldots\ldots\ldots \; (4.7)$$

Substitue (4.3) and (4.7) into equations (4.1) we have for $0 \leq t \leq T$

$$\dot{x}_u(t) = [kx_u(t)]\; u(t); \quad x_u(0) = x_{uo} \;\ldots\ldots\ldots\ldots (4.8a)$$

$$\dot{x}_v(t) = [x_u(t) - x_v(t)]\; v(t); \; x_v(0) = x_{vo} \;\ldots\ldots\ldots (4.8b)$$

in matrix form

$$\dot{\underline{x}} = F \; \underline{x} \; ; \; \underline{x}(0) = \underline{x}_o \qquad \ldots\ldots\ldots\ldots \; (4.9)$$

where

$$\underline{x} = \left[\begin{array}{c} x_u \\ \hline x_v \end{array}\right], \quad F = \left[\begin{array}{c|c} k\; u(t) & 0 \\ \hline v(t) & -v(t) \end{array}\right], \quad \underline{x}_o = \left[\begin{array}{c} x_{uo} \\ \hline x_{vo} \end{array}\right]$$

Note that we have arrived at the same equation as McFarland. However, different rationalizations have been used. The reason for the difference is because it is felt that the assumption $c_u(t)$ and $c_v(t)$ equal to zero in the interval [t, T] used by McFarland later becomes a conflict with the actual values of $C_u(t)$ and $C_v(t)$ in the computation. With the above rationalizations, however, no such assumption has to be made.

Note also that equation (4.9) is linear since neither u(t) nor v(t) is effected by the actual value of the random variables $x_u(t)$ and $x_v(t)$. Therefore, we can say that u(t) and v(t) are not functions of $x_u(t)$ and/or $x_v(t)$. Since $\underline{x}_o$ is Gaussian given by equation (4.2) and $\underline{x}(t)$ is generated by a linear process (4.9), the vector $\underline{x}(t)$ will remain Gaussian.

We now proceed to derive dynamic equations for the set of sufficient statistics of $\underline{x}(t)$. Since $\underline{x}(t)$ has dimension 2, the mean vector will be of dimension 2, and the covariance matrix will contain 3 independent elements. Normally then, the dynamics of this problem should consist of 5 equations. However, it is easy to see that the mean vector is zero:

$$E\,[\underline{x}(t)] = 0 \qquad \forall \quad t \in [0,T] \qquad \ldots\ldots\ldots \text{(4.10)}$$

since the initial value is zero as shown in equation (4.2). The covariance matrix is defined as

$$E\,[\underline{x}(t)\underline{x}^T(t)] = X(t) = \left[\begin{array}{c|c} \overline{x_u^2(t)} & \overline{x_u(t)x_v(t)} \\ \hline \overline{x_u(t)x_v(t)} & \overline{x_v^2(t)} \end{array}\right] \quad \text{...........(4.11)}$$

where $\overline{}$ designates the expected value of the quantity under it.

Now, if $\underline{x}(t)$ satisfy the usual Lipshitz condition then

$$\dot{X}(t) = \frac{d}{dt}\,E[\underline{x}(t)\underline{x}^T(t)] = E\,\frac{d}{dt}\,[\underline{x}(t)\underline{x}^T(t)]$$

$$= E\,[\dot{\underline{x}}(t)\underline{x}^T(t)+\underline{x}(t)\dot{\underline{x}}^T(t)]$$

$$= F\,X + F^T\,X \quad \text{.........(4.12)}$$

Equation (4.12) can be expressed in components as:

$$\frac{d}{dt}\,(\overline{x_u^2}) = k\,u(t)\,\overline{x_u^2(t)} \quad ; \quad \overline{x_u^2}\,(0) = \sigma_u^2\,(0) \qquad (4.13a)$$

$$\frac{d}{dt}\,(\overline{x_u x_v}) = [ku(t)-v(t)]\,\overline{x_u x_v}+v(t)\,\overline{x_u^2}(t)\,;\ \overline{x_u(0)x_v(0)} = 0 \quad \text{...........(4.13b)}$$

$$\frac{d}{dt}\,(\overline{x_v^2}) = -2v(t)\,\overline{x_v^2}(t)+2v(t)\overline{x_u(t)x_v(t)}\,;\ \overline{x_v^2}(0) = \sigma_v^2(0) \quad \text{..(4.13c)}$$

Note that equations (4.13) are <u>nonlinear</u> since the controls u(t) and v(t) are indeed effected by the value of the covariances of the state vector and the products of control and state variables appear.

We can call $\overline{x_u^2}$, $\overline{x_u x_v}$, and $\overline{x_v^2}$ state variables and use equations (4.13) directly as the dynamics or the state equations of the problem. However, as we shall see later, it is more convenient and more meaningful to use the projected intercept and the target miss as the state variables. These variables are defined as

$$x_I(t) \triangleq x_u(t) - x_v(t) \quad \ldots\ldots\ldots\ldots\ldots \quad (4.14a)$$

$$x_T(t) \triangleq kx_u(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.14b)$$

where

$x_I(t)$ = current value of target miss

$x_T(t)$ = current value of projected intercept

then

$$\dot{x}_I(t) = u(t)\, x_T(t) - v(t)\, x_I(t) \qquad (4.15a)$$

$$\dot{x}_T(t) = k\, u(t)\, x_T(t) \qquad (4.15b)$$

Again, it is easy to see from (4.14) that the mean values of $x_I(t)$ and $x_T(t)$ remains zero throughout the interval [o,T]. Define the covariance matrix as

$$E \begin{bmatrix} x_I \\ \hline x_T \end{bmatrix} \cdot \begin{bmatrix} x_I & x_T \end{bmatrix} = \left[\begin{array}{c|c} \overline{x_I^2} & \overline{x_I x_T} \\ \hline \overline{x_I x_T} & \overline{x_T^2} \end{array}\right] = \left[\begin{array}{c|c} x_o & x_1 \\ \hline x_1 & x_2 \end{array}\right] \qquad (4.16)$$

and again by the process similar to equation (4.12) we can show that the elements of the covariance matrix satisfy the following differential equations:

$$\dot{x}_o(t) = -2v(t)x_o(t) + 2u(t)x_1(t); \quad x_o(0) = \sigma_u^2(0)+\sigma_v^2(0) \qquad (4.17a)$$

$$\dot{x}_1(t) = \left[ku(t)-v(t)\right]\cdot x(t)+u(t)x_2(t); \quad x_1(0) = k\,\sigma_u^2(0) \qquad (4.17b)$$

$$\dot{x}_2(t) = 2k\ u(t)\ x_2(t) \qquad ; \quad x_2(0) = k^2 \sigma_u^2(0) \quad ..(4.17c)$$

Equations (4.17) generates a set of sufficient statistics for the problem since $\bar{x}_I(t)$ and $\bar{x}(t)$ are zero throughout the interval [0, T]. These equations then will serve as dynamics or state equations for our problem. A successful use of these equations in solving the problem will demonstrate that a stochastic differential game can be treated as a deterministic game if a set of sufficient statistics can be found and used as state variables in the modelling of the state equations.

4.2.2 Cost Function of the Problem

In order to find the "best" controls for each player, some criteria will have to be established to discriminate one control from another. Since one of our goal is to try to be realistic as possible, and since this is a stochastic problem, the probability of survival of the target seems to be the ultimate criteria. The attacking missle, wanting to destroy the target, will try to minimize the probability of survival of the target; whereas the interceptor, defending the target, will try to maximize the probability of survival of the target. We shall now attempt to find the probability of survival as a function of the state variables.

$$\begin{aligned} P(\text{survival}) &= 1 - P(\text{not captured and score}) \quad ..(4.18) \\ &= 1 - P(\text{score/not captured})P(\text{not captured}) \end{aligned}$$

The last step follows from the Baysean's Law of conditional probability. Using the definition of the scoring zone,

$$P(\text{score/not captured}, x_T(T)) = \begin{cases} 1 & \text{for} \quad -r_s < x_T(T) < r_s \\ 0 & \text{otherwise} \end{cases} \qquad \ldots\ldots(4.19)$$

Now

$$P(\text{score}, x_T(T)/\text{not captured}) = P(\text{score}/x_T(T), \text{not captured}) \cdot P(x_T(T)/\text{not captured})$$

Therefore

$$P(\text{score/not captured}) = \int_{-r_s}^{r_s} f_{x_T(T)/\text{not captured}}(\xi)\,d\xi \qquad \ldots\ldots\ldots(4.20)$$

where the conditional probability density function is defined as

$$f_{x_T(T)/\text{not captured}}(\xi) = \frac{1}{\sqrt{2\pi x_2(T)}} \exp\left[-\xi^2/2x_2(T)\right]$$

substitue this into (4.20) and use the symmetric property of the normal probability density function

$$P(\text{score/not captured}) = 2\int_0^{r_s} \frac{1}{\sqrt{2\pi x_2(T)}} \exp\left[-\xi^2/2x_2(T)\right] d\xi$$

$$= \operatorname{erf}\left[\frac{r_s}{\sqrt{2x_2(T)}}\right] \qquad \ldots\ldots\ldots(4.21)$$

where $$\operatorname{erf}\left[\frac{r_s}{\sqrt{2x_2(T)}}\right] \triangleq \frac{2}{\sqrt{\pi}} \int_0^{r_s/\sqrt{2x_2(T)}} \exp(-\varsigma^2)\,d\varsigma \qquad \ldots\ldots\ldots(4.22)$$

The error function (erf) is a Fortran built-in function and can be called directly from the computer using Fortran language. Now

$$P(\text{not captured}) = 1 - P(\text{captured})$$

Using the definition of the capture zone

$$P(\text{captured}/x_I(T)) = \begin{cases} 1 & \text{if } -r_c < x_I(T) < r_c \\ 0 & \text{otherwise} \end{cases} \qquad \ldots\ldots\ldots (4.23)$$

Thus

$$P(\text{not captured}) = 1 - \int_{-r_c}^{r_c} f_{x_I(T)}(\xi)\,d\xi \ldots\ldots\ldots (4.24)$$

since $x_I(T)$ is Gaussian, we can show that

$$P(\text{not captured}) = 1 - \text{erf}\left[\frac{r_c}{\sqrt{2x_o(T)}}\right] \qquad \ldots\ldots\ldots (4.25)$$

substitue (4.2]) and (4.25) back into (4.18) we obtain

$$\begin{aligned} \phi\,[x_o(T),\ x_2(T)] &= P(\text{survival}) \\ &= 1-\text{erf}\left[\frac{r_s}{\sqrt{2x_2(T)}}\right]\left\{1-\text{erf}\left[\frac{r_e}{\sqrt{2x_o(T)}}\right]\right\} \end{aligned}$$

$$\ldots\ldots\ldots (4.26)$$

By definition, both $x_o(T)$ and $x_2(T)$ must be positive. From (4.26) when $x_o(T) = 0$ then P(survival) = 1. This checks out with the fact that perfect interception ensures certain survival. For combinations of low $x_2(T)$ and high $x_o(T)$ the probability of survival approaches zero, this again checks out with the condition when the attacking missle successfully evaded the interceptor and yet manages to reach the target.

In order to realize a reasonable and realistic controls for the problem, and integral penalty function must be added to the cost function. The most direct method is to penalize the squared values of the controls u(t) and v(t) with proper weights. Using this approach, we define

$$I(u,v) \triangleq \int_o^T [Q_1 u^2(t) - Q_2 v^2(t)]dt \quad ....(4.27)$$

where Q_1 and Q_2 are positive quantities representing appropriate penalizing weights on controls u(t) and v(t) respectively. The actual choice of Q_1 and Q_2 will be discussed later. The penalizing term for U is positive because U is trying to minimize the cost function. This term will restrict U from using "too large" control. Player V has a negative penalizing term because he is trying to maximize the cost function. Too large v(t) in any interval of time could results in a negative value for I(u,v). The composite cost functional for this problem will then be:

$$J(u,v) = \phi[x_o(T), x_2(T)] + I(u,v) \quad .... (4.28)$$

4.2.3 Constraints

This pursuit-evasion problem has been formulated in such a way that the subsequent effective lateral vel. ities will not exceed 10% of the associated average forward speeds. Thus the assumption that the forward motion is uncoupled from the lateral motion can be used throughout this chapter. In addition, hard constraints are put on u(t) and v(t) as follows:

$$|u(t)| \leqslant 2, \quad |v(t)| \leqslant 1 \qquad \forall\ t \in [o,T_s] \quad ....(4.29)$$

These constraints have an equivalent effects of limiting the lateral accelerations of the missles. The larger limiting factor for U as compared to V is used in order to be consistent with other parameters which will be discussed in the section on the computational aspects of the problem.

4.3 Convergence Control Technique

Before we embark on other computational aspects of the problems, it is well to note here that the DDP algorithm will not converge for this problem without the use of some kind of convergence control scheme. McFarland need the "step-size" method developed by Jacobson and Mayne in solving his problem with good result. The so called "step-size" method when used with the first-order algorithm developed in this report, however, did not warrant convergence for this pursuit-evasion problem.

Obviously, some other convergence control scheme is required. One such scheme which has demonstrated good convergence property for a large varieties of problems was developed by Jarmark in 1975. Anderson[43] has used this scheme to derive feedback control for pursuing spacecraft with excellent results. After the "step-size" method failed to provide convergence for the solution of this problem, several other convergence control schemes were tried. It was finally decided that Jarmark's scheme was the most suitable for this problem. This scheme will be briefly described in the rest of this section. More detailed discussion and proof can be found in references (32) to (34).

The reason why the actual cost change deviates too much from the predicted cost change is the violation of the assumption that $\Delta \underline{x}$ is small in the derivation of the DDP equations and the higher order terms in equation (2.33) cannot be neglected. Since the DDP algorithm worked out in chapter 2 deals with the inner minimization, we shall also deal exclusively only with DDP minimization here. However, it is clear that the same technique can be used with inner maximization for the minmax case also with only a few minor adjustments.

Jarmark has shown[34] that the magnitude of $\Delta \underline{x}(t)$ can be restricted and the Taylor's series expansion equation (2.34) can be made valid by adding a penalty term to the integral of the cost functional equation (2.20). Thus equation (2.20) can be rewritten as follows:

$$J(\underline{u}(t)) = \phi\,(\underline{x}(T),T) + \int_0^T [L(\underline{x},u,t) + \Delta \underline{u}(t)^T W \Delta \underline{u}(t)]dt \qquad \text{........... (4.30)}$$

Then Jarmark proceeds to show by using Theorems and Lemmas that:

1. $\Delta \underline{u}$ in each iteration as measured by the metric

$$d(\underline{u}^i, \underline{u}^{i-1}) = \int_0^T \| \underline{u}^i - \underline{u}^{i-1} \| \; dt \qquad \text{........(4.31)}$$

can be made arbitrarily small by the choice of the weighting matrix W.

2. There exists a W such that the series expansion equation (2.33) and (2.34) is valid.

3. For $W > 0$, a reduction in cost at each iteration is obtained if $\Delta \underline{u}(t) \neq 0$ for some $t \in [0,T]$.

4. The solution of the artificial cost in equation (4.30) converges to the same solution of the original cost equation (2.20).

These are existence Theorems, and so far there is no hard and fast rule on how to choose W. If the element of W is too large the convergence will be slow. On the other hand, if the element of W is too small, the assumption $\Delta \underline{x}$ is small may not be valid. Jarmark suggests the following procedure.

For a starting value choose a W base on prior experience on the same type of problems. The structure of the problem could be used, for example, the elements of W should be small when the problem is close to a linear problem. After each iteration, the stopping rule will have to be changed from $|a(0)| < \varepsilon$ to

$$|a(0)| < \frac{\varepsilon}{1 + \|W\|} \qquad \text{........(4.31)}$$

If the stopping rule is not satisfied then use the convergence index domain shown in Figure 7 to adjust the element of W.

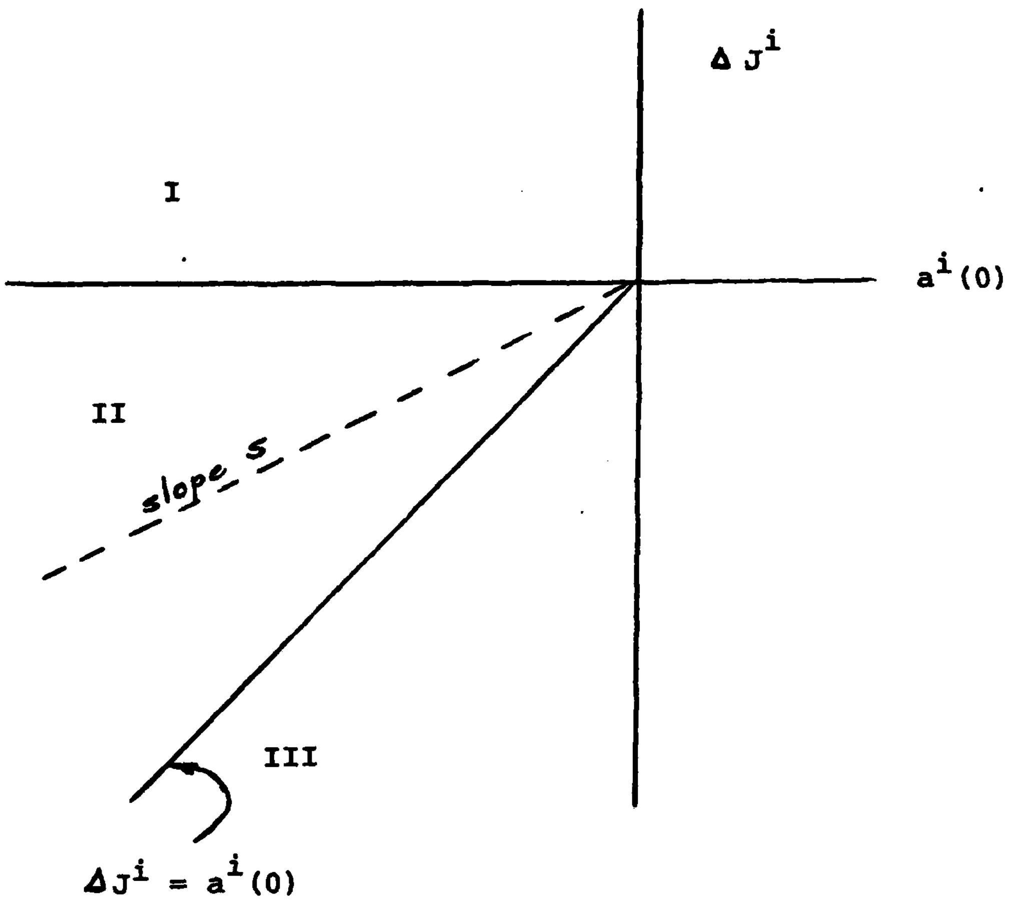


Figure 7. Convergence index domain

Area I: $\frac{\Delta J^i}{a^i(0)} < 0$, do not except the iteration, increase the component 2-5 times.

Area II: the element w^i of W can be adjusted by the following formula

$$w^i = (1 - \Delta J^i/(a^i(0)s))(H^i_{uu} + w^i) \quad \text{.........}(4.33)$$

Area III: use approximately the same value of w^i in the last iteration. w^i may be increased or decreased slightly in this situation. Increase if close to ΔJ^i axis, and decrease otherwise.

This procedure is used with very good convergence property for the present problem.

4.4 Computational Aspects of the Problem

We now have the problem in which the attacking missle U has to find $\hat{u}(t)$ to minimize the maximum possible cost and the intercepting missle V has to find v(t) to maximize the minimum possible cost. The cost functional and the dynamics of the problem as developed in section 4.2 may be written as follows:

Cost Functional

$$J(u,v) = 1\text{-erf}\left[\frac{r_s}{\sqrt{2x_2(T)}}\right] \cdot \left\{ 1\text{-erf}\left[\frac{r_c}{\sqrt{2x_o(T)}}\right]\right\} + \int_0^T [Q_1u^2(t) - Q_2v^2(t)]dt \quad \ldots\ldots\ldots(4.34)$$

Dynamics

$$\dot{x}_o = -2vx_o + ux_1 \quad ; \quad x_o(0) = \sigma_u^2(0) + \sigma_v^2(0) \quad \ldots\ldots\ldots(4.35)$$

$$\dot{x}_1 = [ku - v]\, x_1 + ux_2 \quad ; \quad x_1(0) = k\,\sigma_u^2(0) \quad \ldots\ldots(4.35b)$$

$$\dot{x}_2 = 2kux_2 \quad ; \quad x_2(0) = k^2\,\sigma_u^2(0) \ldots\ldots(4.35c)$$

These state equations are valid for $t \in [0,T]$

Constraints

$$|u(t)| \leqslant 2, \quad |v(t)| \leqslant 1 \quad \text{for} \quad 0 \leqslant t \leqslant T \quad \ldots\ldots.(4.36)$$

4.4.1 Parameter Value Assignment

The missles engagement distance is assumed typically around 35 miles or L = 180 kilofeet. The kilofeet unit is used here because it is more convenient and more widely used unit for this type of problem. As mentioned before the forward motion is uncoupled to the lateral motion. Therefore, only the average forward speeds for the players rather than the instantaneous forward speeds in the whole engagement time interval are needed. Typical forward speed for the attacking missle is S_u = 15 kilofeet/second while the intercepting missle is typically slower at S_v = 7.5 kilofeet per second.

At these average speed, the players will cross the line of interception at the time:

$$T = \frac{L}{S_u + S_v} = 8 \text{ seconds}$$

The attacking missle, if escaped from the interceptor, will cross the scoring boundary at the time:

$$T_S = \frac{L}{S_u} = 12 \text{ seconds}$$

Using these average forward speeds, the distance between the line of interception and the target is:

$$L_I = S_v T = 60 \text{ kilofeet}$$

The initial lateral dispersions from the line L are normally distributed with the mean zero for both players. The initial lateral position of the attacking missle is simply more dispersed because it involved more distance and time covered before commencement of the differential game. The initial standard deviation for U's lateral position is $\sigma_u(0) = 3$ kilofeet while that of V is $\sigma_v(0) = 0.5$ kilofeet.

The fact that the scoring zone is larger than the capture zone should be clear and has been explained in the description of the problem. We shall assume $r_s = 0.5$ kilofeet and $r_c = 0.25$ kilofeet for the purpose of this study.

A typical way of selecting the weights Q_1 and Q_2 for the penalty functions of the controls is to use

$$Q_1^{-1} = T \times \text{maximum value of } u^2 = 32$$

$$Q_2^{-1} = T \times \text{maximum value of } v^2 = 8$$

Experimentation around these values gives $Q_1 = 0.0625$ and $Q_2 = 0.125$ for the best results in this study. McFarland also used these values in his report. Between the time of interception and the target boundary crossing time, we have shown that U can cut his lateral dispersion down by a fixed fraction k depending upon the other parameter values. We shall assume $k = 0.5$ for this report.

In summary, we shall use:

$T = 8$ sec $\quad \sigma_u(0) = 3$ kft $\quad r_s = 0.5$ kft $\quad Q_1 = 0.0625$

$k = 0.5 \quad \sigma_v(0) = 0.5$ kft $\quad r_c = 0.25$ kft $\quad Q_2 = 0.125$

4.4.2 Maxmin Solution

We shall follow the algorithm steps covered in section 2.5. To start off the algorithm for the maxmin control a nonimal control $v_o(t)$ can be approximated by maximizing the cost functional equation (4.34) subject to the dynamic equations (4.35) with control $u(t) = 0 \; \forall t \in [0,T]$. Using this nominal control for V will have the effect of forcing U to do "something" in order to minimize the cost functional.

With $u(t) = 0$, the state equations (4.35) become

$$\dot{x}_o = -2vx_o \quad ; \quad x_o(0) = \sigma_u^2(0) + \sigma_v^2(0) \quad \ldots\ldots.(4.37a)$$

$$\dot{x}_1 = -vx_1 \quad ; \quad x_1(C) = k\,\sigma_u^2(0) \quad \ldots\ldots.(4.37b)$$

$$\dot{x}_2 = 2kux_2 \quad ; \quad x_2(0) = k^2\,\sigma_u^2(0) \quad \ldots\ldots.(4.37c)$$

With these state equations, the Hamiltonian is:

$$H = Q_1u^2 - Q_2v^2 - 2vx_o\,\lambda_o - vx_1\,\lambda_1 \quad \ldots\ldots.(4.38)$$

where λ's are costate variables expressed by the following differential equations:

$$\dot{\lambda}_o = -\frac{\partial H}{\partial x_o} = xv\,\lambda_o \quad ; \lambda_o(T) = \frac{\partial \phi}{\partial x_o(T)} \quad \ldots.(4.39a)$$

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = v\,\lambda_1 \quad ; \lambda_1(T) = 0 \quad \ldots.(4.39b)$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} = 0 \quad ; \lambda_2(T) = \frac{\partial \phi}{\partial x_2(T)} \quad \ldots.(4.39c)$$

Assuming for the moment that $|v(t)| < 1$ for $0 \leqslant t \leqslant T$, the optimality condition is:

$$\frac{\partial H}{\partial v} = -2Q_2 v - 2x_o \lambda_o - x_1 \lambda_1 = 0$$

Hence

$$v(t) = -(2x_o \lambda_o + x_1 \lambda_1)/2Q_2 \qquad \text{..........(4.40)}$$

Differentiating (4.40)

$$\dot{v}(t) = -(2\dot{x}_o \lambda_o + 2x_o \dot{\lambda}_o + \dot{x}_1 \lambda_1 + x\dot{\lambda}_1)/2Q_2 \text{..(4.41)}$$

Using (4.37) and (4.39) in (4.41) we have

$$\dot{v}(t) = -(-2vx_o \lambda_o + 2vx_o \lambda_o - vx_1\lambda_1 + vx_1\lambda_1)/2Q_2 \doteq 0 \qquad (4.42)$$

Therefore, $v(t) = \text{constant}$ .... (4.43)

Now, equations (4.37) and (4.39) can be solved analytically with the following results:

$$\begin{aligned} x_o(t) &= x_o(0)e^{-2vt} &, \quad \lambda_o(t) &= \lambda_o(T) \\ x_1(t) &= x_1(0)e^{-vt} &, \quad \lambda_1(t) &= 0 \\ x_2(t) &= x_2(0) &, \quad \lambda_2(t) &= \lambda_2(T)e^{2v(t-T)} \end{aligned}$$

.... (4.44)

Substitue (4.44) back into (4.40)

$$v(t) = -2x_o(0)\, \lambda_o(T)e^{-2vt} \qquad \text{..............(4.45)}$$

and from (4.39a) we have

$$\lambda_o(T) = -\frac{r_c}{(2x_o(T))^{3/2}} \cdot \left[\frac{2}{\sqrt{\pi}} e^{-r_c^2/2x_o(T)} \right] . \mathrm{erf} \left[\frac{r_s}{\sqrt{2}\ k\ \sigma_u(0)} \right] \qquad \text{.............(4.46)}$$

Use $x_o(T) = e^{-2vT}\left[\sigma_u^2(0) + \sigma_v^2(0)\right]$ in (4.46) and substituting back into (4.45) yeilds

$$v = \frac{r_c}{Q_2\sqrt{2\pi(\sigma_u^2(0) + \sigma_v^2(0))}} . \exp[vT - r_c^2 \exp(2vT)/2(\sigma_u^2(0)+\sigma_v^2(0))] . \mathrm{erf}\left[\frac{r_s}{\sqrt{2}\ k\ \sigma_u(0)}\right] \qquad \text{........(4.47)}$$

Notice that equation (4.47) is transcendental, it must be solved numerically. With the parameters given in section (4.4.1), we found that $v \approx .25$. Therefore, we shall use $v_o(t) = .25$ for $0 \leqslant t \leqslant T$ as the starting nominal maxmin control for our algorithm.

The next step is to find all the local minima for $J(u,v_o)$. Two local minimizing $u_o^{(1)}$ and $u_o^{(2)}$ are found by repeated applications of DDP routine for different values of starting $u(t)$. Table 10 summarizes the numerical results for the maxmin iterations using $v_o(t) = .25$, $u_o^{(1)}(t) = -.25$, and $u_o^{(2)}(t) = .25$. The two starting values of $u(t)$ lead to two different minima. Extensive preliminary testing shows that only these two local minima exist for this problem.

The pertinent equations for the applications of DDP routine are as follows:

$$H = Q_1u^2 - Q_2v^2 + 2(ux_1-vx_o)J_{x_o} + ((ku-v)x_1+ux_2)J_{x_1} + 2kux_2J_{x_2} \quad \text{.........(4.48)}$$

$$u^*(t) = -(2x_1J_{x_o}+(kx_1+x_2)J_{x_1}+2kx_2J_{x_2}-2Wu_o)/2(Q_1+W) \quad \text{....(4.49)}$$

If $|u^*(t)| > 2$, then set the corresponding $|u^*(t)| = 2$ since H is convex with respect to u. If $u^*(t)$ is not on the boundary then equations (2.49) become

$$\dot{a}(t) = Q_1(u^*(t) - u_o(t))^2 \quad \text{.......... (4.50a)}$$

$$\dot{J}_{x_o}(t) = 2v_o(t)\ J_{x_o}(t) \quad \text{.......... (4.50b)}$$

$$\dot{J}_{x_1}(t) = -2u^*(t)J_{x_o}(t) - (ku^*(t)-v_o(t))J_x(t) \quad \text{......(4.50c)}$$

$$\dot{J}_{x_2}(t) = -u^*(t)\ [J_{x_1}(t) + 2k\ J_{x_2}(t)] \quad \text{........(4.50d)}$$

with the terminal conditions

$$a(T) = 0 \quad \text{...........(4.51a)}$$

$$J_{x_o}(T) = -r_c\exp(-r_c^2/2x_o(T)\mathrm{erf}\ (r_s/\sqrt{2x_2(T)})/\sqrt{2\pi}\ x_o(T)^{3/2} \quad \text{...........(4.51b)}$$

$$J_{x_1}(T) = 0 \quad \text{...........(4.51c)}$$

$$J_{x_2}(T) = r_s\ \exp(-r_s^2/2x_2(T)).\ [1-\mathrm{erf}(r_c/\sqrt{2x_o(T)})]/\sqrt{2\pi}\ x_2(T)^{3/2} \quad \text{...........(4.51d)}$$

Table 10. Maxmin Iteration

MAX	MIN	$J^{(1)}$	W	a(0)	ΔJ	MAX	MIN	$J^{(2)}$	W	a(0)	ΔJ
v_o	$u_o^{(1)}$	.712	1.00	-.013	-.194	v_o	$u_o^{(2)}$	.878	1.00	-5.0E-4	-1.8E-2
		.518	0.90	-.0021	-.068			.860	0.90	-5.8E-4	-1.9E-2
		.450	0.81	-.0016	-.047			.840	0.81	-6.6E-4	-2.0E-2
		.403	0.73	-.001	-.025			.821	0.73	-7.1E-4	-1.9E-2
		.378	0.66	-5.1E-4	-.011			.802	0.66	-6.8E-4	-1.5E-2
		.367	0.59	-2.3E-4	-4E-3			.787	0.59	-5.0E-4	-1.0E-2
		.363	0.53	-1.3E-4	-2E-3			.778	0.53	-3.6E-4	-7E-3
		.361	0.48	-9.0E-5	-1E-3			.771	0.48	-3.6E-4	-6E-3
	$\hat{u}_o^{(1)}$	.360	0.43					.764	0.43	-3.5E-4	-6E-3
								.764	0.43	-3.5E-4	-6E-3
								.759	0.39	-3.4E-4	-5E-3
								.754	0.35	-3.1E-4	-4E-3
								.749	0.31	-2.9E-4	-3E-3
								.746	0.28	-2.5E-4	-3E-3
								.743	0.25	-2.8E-4	-2E-3
								.741	0.41	-1.6E-4	-1E-3
								.739	0.67	-7.0E-5	-1E-3
							$\hat{u}_o^{(2)}$	.738	0.59		
			$(g_1^{(1)}, g_1^{(1)})$ = .124, STEP = 0.58					$(g_1^{(1)}, g_1^{(2)})$ = .0094			

Table 10.(Continue)

MAX	MIN	$J^{(1)}$	W	a(0)	ΔJ	MAX	MIN	$J^{(2)}$	W	a(0)	ΔJ
v_1	$u_1^{(1)}$	.442	1.00	-9E-5	-2E-3	v_1	$u_1^{(2)}$	.740	0.73	-2.3E-4	-4E-3
		.440	0.90	-7E-5	-2E-3		$u_1^{(2)}$	.736	0.66		
		.438	0.81	-6E-5	-1E-3						
	$\hat{u}_1^{(1)}$	.436	0.73								
$(g_2^{(1)}, g_2^{(1)}) = .248$, STEP = 0.35 $(g^{(1)}, g^{(2)}) = -.02$											
v_2	$u_2^{(1)}$	.564	1.00	-3.5E-4	-9E-3	v_2	$u_2^{(2)}$	.720	0.54	-7.4E-4	-1.1E-2
		.555	0.90	-2.7E-4	-6E-3		$u_2^{(2)}$	.709	0.48	-1.4E-4	-2 E-3
		.548	0.81	-2.1E-4	-5E-3			.707	0.43		
		.544	0.73	-1.6E-4	-3E-3						
		.541	0.66	-1.2E-4	-3E-3						
		.538	0.59	-9E-5	-1E-3						
	$\hat{u}_2^{(1)}$	.537	0.54								
$(g_3^{(1)}, g_3^{(1)}) = .788$, STEP = 0.14 $(g_3^{(1)}, g_3^{(2)}) = -0.24$											
v_3	$u_3^{(1)}$	.706	1.00	-5.8E-4	-1.4E-2	v_3	$u_3^{(2)}$	.678	0.48	-1.1E-3	-1.7E-2
		.692	0.90	-4.4E-4	-1.0E-2			.662	0.43	-3.1E-4	-4.0E-3
		.682	0.81	-3.2E-4	-6.0E-3			.658	0.39		
		.676	0.73	-2.3E-4	-4.0E-3						
		.672	0.66	-1.7E-4	-3.0E-3						
		.669	0.59	-1.2E-4	-2.0E-3						
		.667	0.53	-0.0E-5	-1.0E-3						
	$\hat{u}_3^{(1)}$	.666	0.48								
MINCOST CROSSOVER											

The time T = 8 seconds is divided into 64 intervals and equation (4.49) is used before each step of integrating equations (4.50) backward, the value of $u^*(t)$ for each step is also stored in the memory to be used either as the minimizing control or as the nominal control $u_o(t)$ for the next iteration. Equation (4.50a) is used in case $u^*(t)$ is not on the boundary. If $u^*(t)$ is on the boundary, however, the following equation must be used

$$-\dot{a}(t) = H(\underline{x}, u^*, J_{\underline{x}}, t) - H(\underline{x}, u_o, J_{\underline{x}}, t) \quad \ldots\ldots\ldots\ldots (4.52)$$

with the same terminal condition equation (4.51a).

It must be noted that the simple Euler integration scheme used very effectively in the last chapter is not an adequate integration scheme for both the state and the DDP equations. More accurate integration scheme was needed, one such scheme is the Runge-Kutta fourth order integration method. The Runge-Kutta integration scheme was used both in forward integration of the state equations (4.17) and the backward integration of DDP equations (4.50).

Refer to Table 10, for $v_o(t) = 0.25$ and $u_o^{(1)}(t) = -0.25$, the cost is $J^{(1)} = .712$. The convergence control weighting factor W = 1. After integrating (4.50) back to t = 0, the predicted cost change a(0) = -.013. Using the new control, $u^*(t)$ found in the process of backward integration, the new cost was evaluated and the cost change $\Delta J = -.194$. This process was repeated until a(0) is smaller than $\frac{.001}{1+W}$.

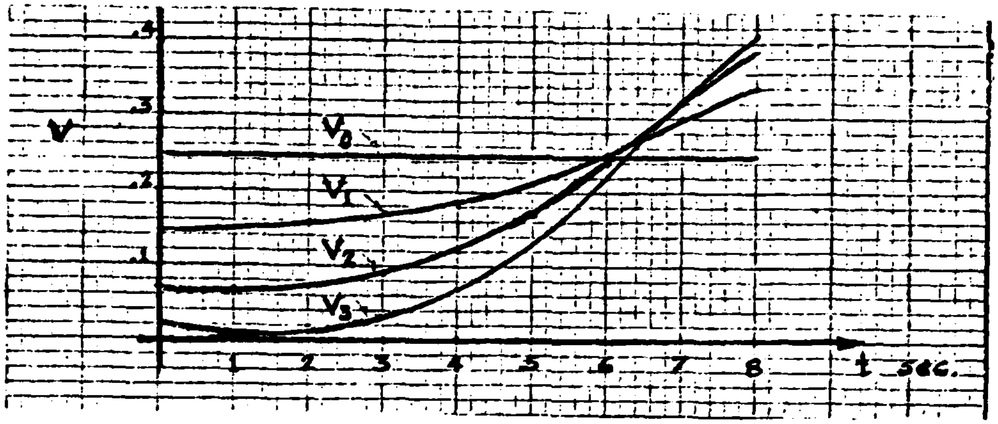


Figure 8a. Successive Approximations to the Maxmin Control

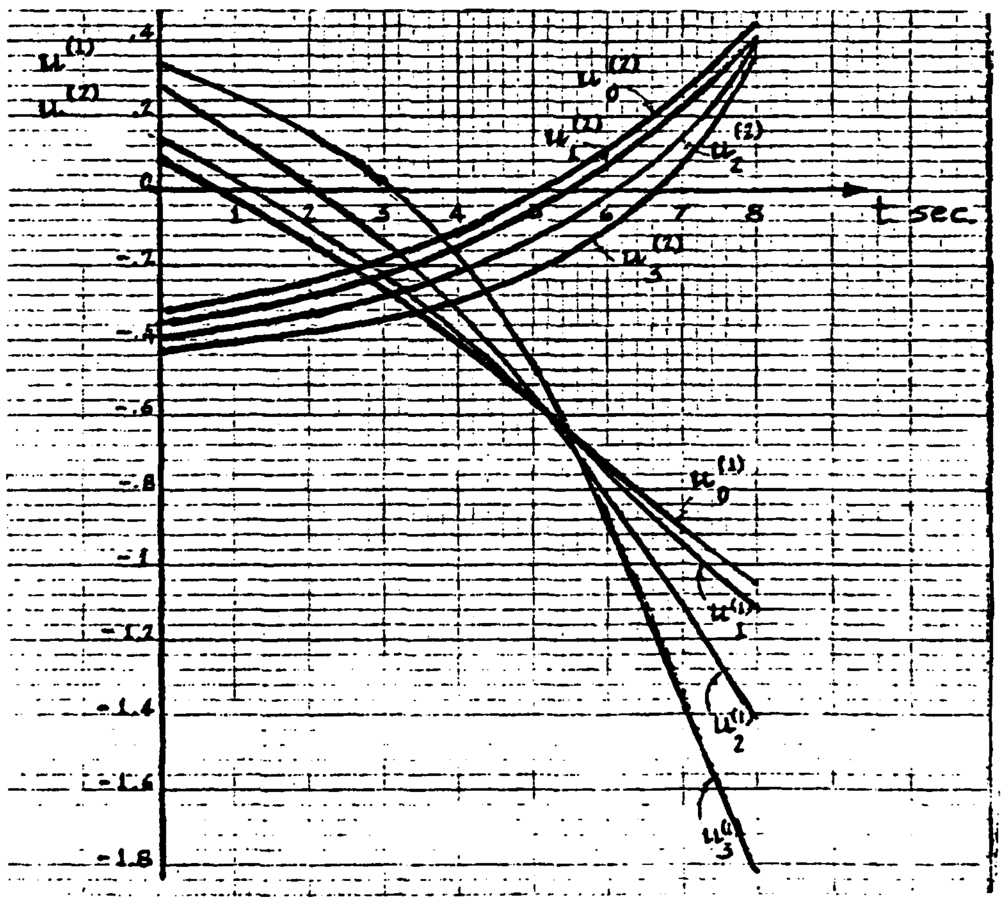


Figure 8b. Locally-Minimizing u(t) for each Successive Approximation

MAXMIN SOLUTION FOUND COST1= 0.55904 COST2= 0.55904

TIME	USTAR1	USTAR2	VSTAR
1	0.30974	-0.43563	0.02497
2	0.29594	-0.43200	0.02333
3	0.28530	-0.42847	0.02181
4	0.27780	-0.42502	0.02040
5	0.26743	-0.42163	0.01913
6	0.25715	-0.41830	0.01800
7	0.24695	-0.41502	0.01702
8	0.23675	-0.41175	0.01620
9	0.22663	-0.40851	0.01555
10	0.21644	-0.40526	0.01509
11	0.20617	-0.40199	0.01483
12	0.19577	-0.39869	0.01476
13	0.18518	-0.39536	0.01492
14	0.17435	-0.39196	0.01530
15	0.16322	-0.38849	0.01593
16	0.15171	-0.38493	0.01680
17	0.13977	-0.38127	0.01793
18	0.12730	-0.37750	0.01934
19	0.11425	-0.37359	0.02103
20	0.10052	-0.36954	0.02302
21	0.08603	-0.36533	0.02531
22	0.07070	-0.36093	0.02791
23	0.05444	-0.35635	0.03083
24	0.03716	-0.35155	0.03409
25	0.01878	-0.34653	0.03769
26	-0.00079	-0.34125	0.04164
27	-0.02164	-0.33572	0.04594
28	-0.04386	-0.32990	0.05060
29	-0.06753	-0.32378	0.05563
30	-0.09274	-0.31734	0.06103
31	-0.11955	-0.31055	0.06681
32	-0.14804	-0.30340	0.07296
33	-0.17828	-0.29585	0.07949
34	-0.21034	-0.28788	0.08640
35	-0.24425	-0.27947	0.09368
36	-0.28009	-0.27058	0.10134
37	-0.31787	-0.26119	0.10937
38	-0.35764	-0.25125	0.11777
39	-0.39940	-0.24074	0.12652
40	-0.44317	-0.22961	0.13563
41	-0.48894	-0.21782	0.14507
42	-0.53668	-0.20532	0.15485
43	-0.58637	-0.19207	0.16494
44	-0.63796	-0.17801	0.17533
45	-0.69138	-0.16307	0.18601
46	-0.74656	-0.14720	0.19696
47	-0.80339	-0.13032	0.20815
48	-0.86177	-0.11236	0.21958
49	-0.92156	-0.09322	0.23122
50	-0.98263	-0.07282	0.24304
51	-1.04481	-0.05105	0.25502
52	-1.10793	-0.02779	0.26714
53	-1.17179	-0.00292	0.27937
54	-1.23620	0.02370	0.29169
55	-1.30095	0.05222	0.30406
56	-1.36582	0.08282	0.31646
57	-1.43059	0.11567	0.32885
58	-1.49504	0.15099	0.34122
59	-1.55894	0.18900	0.35353
60	-1.62206	0.22996	0.36576
61	-1.68419	0.27413	0.37787
62	-1.74513	0.32182	0.38984
63	-1.80467	0.37334	0.40165
64	-1.86265	0.42906	0.41326

Table 11. Computer Printout of Maxmin Solution

After eight iterations, the DDP routine converged, the latest nominal control is the local minimizing control, $u_o^{(1)}(t)$ plotted in Figure 8b.

The second local minimizing control, $u_o^{(2)}(t)$ also shown in Figure 8b was found in a similar manner using $v_o(t) = 0.25$ and $u_o^{(2)}(t) = -0.25$ as starting nominal controls. With these controls, $J^{(2)} = .878$. Again using W = 1 the predicted cost change was found to be -.0005 while the actual cost change was -.018, the minimizing control in this iteration was accepted as the new nominal control and so on. DDP routine in this case converged in seventeen iterations.

At the two local minimizing controls, the function - space gradient of mincost with respect to the maximizing control is found by the following equation:

$$g(t) = -2Q_2 v(t) - 2x_o(t)\, J_{x_o}(t) - x_1(t) J_{x_1}(t) \quad \ldots\ldots\ (4.53)$$

The norm $(g^{(1)}, g^{(1)})$ and the inner product $(g^{(1)}, g^{(2)})$ was found by using

$$(g^{(i)}, g^{(j)}) = \int_0^T g^i(t)\, g^j(t)\,dt \quad \ldots\ldots\ldots\ (4.54)$$

Using the gradient: $g_1^{(1)}(t)$ and $g_1^{(2)}(t)$, the norm: $(g_1^{(1)}(t), g_1^{(1)}(t))$, and the inner-product: $(g_1^{(1)}(t), g_1^{(2)}(t))$, the step length was calculated. The logic used to find the step length for this problem is to alter the maximizing control $v_o(t)$ in such a way that appreciable increase of the mincost $J^{(1)}$ is found and yet the minimizing control use in the last iteration will not be too far off from the new one. In addition, we do not want the new mincost $J^{(1)}$ to be greater than the new mincost $J^{(2)}$. Experimentation shows that the step chosen to realize a predicted change in mincost of 20% worked very well in most cases. However, the step size limited to a maximum value of one. The new approximation to the maxmin control is then found by

$$v_1(t) \quad = \quad v_o(t) + \text{step} \;.\; g_2^{(1)}(t) \ldots\ldots\ldots\ldots\ldots \quad (4.54)$$

This is plotted in Figure 8a.

The first new local min for $v_1(t)$ was reached in three iterations of the DDP routine, while the second one was reached in one iterations. Examination of the mincosts reveals that they are approaching one another. Thus we may suspect that a crossover point may be found. After three outer approximations to the maxmin control, the crossover point was indeed found, the value of the controls at this crossover point is shown in Table 11.

In figure 9, pertinent state information are presented for the control combination $\tilde{v}(t)$ and $\tilde{u}^{(1)}(t)$ or the maxmin control combination number 1. The standard deviation for the Target Miss is plotted as $\sqrt{x_2}^{(1)}$; this curve shows that the probability density function of the projected Target Miss first expands because of the positive value of $\tilde{u}^{(1)}(t)$ up till the time t = 3.25 seconds, then the negative

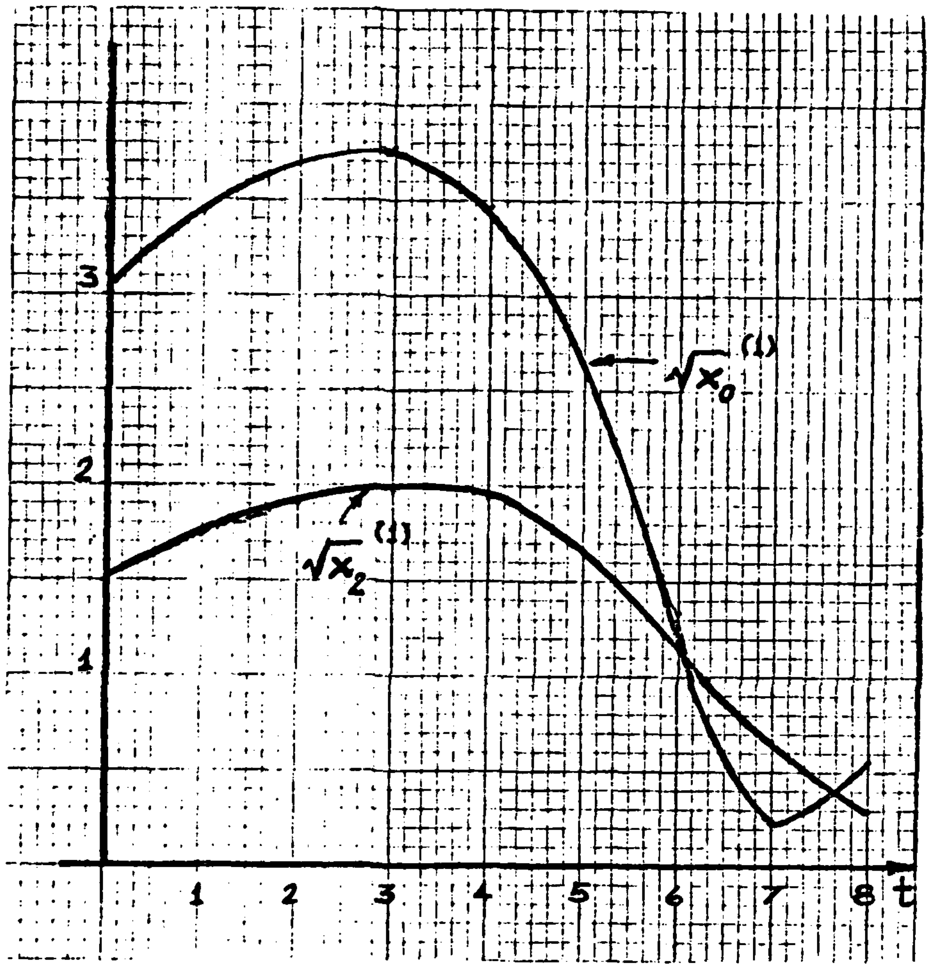


Figure 9. Maxmin Standard Deviation for Projected Intercept and Target Miss, Minimum No. 1

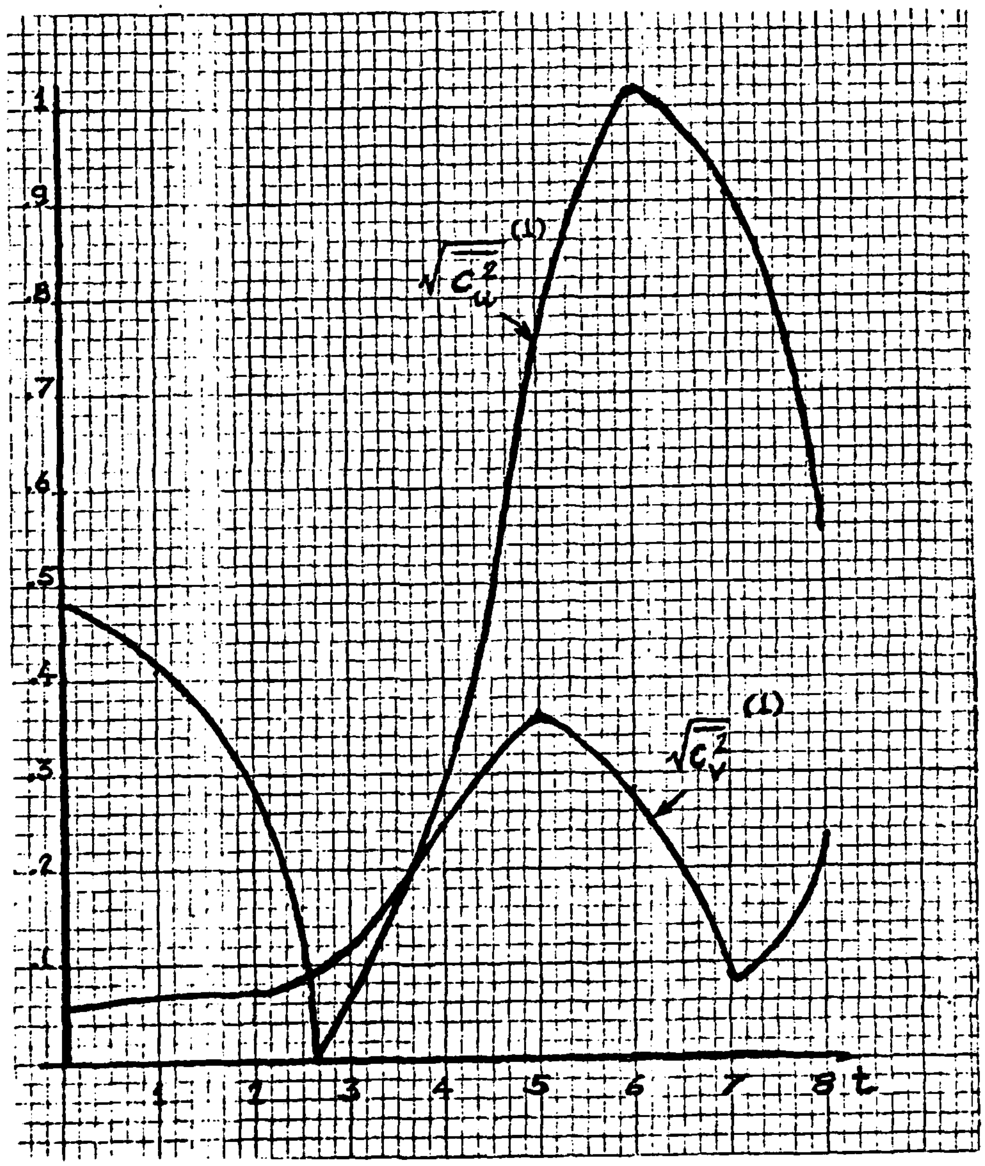


Figure 10. Maxmin Effective Lateral Velocities, Minimum No. 1

value of $\tilde{u}^{(1)}(t)$ causes the same process probability density function to continually contract and end up with $\sqrt{x_2^{(1)}}$ = 0.3 kilofeet, a significant drop from the starting value of 1.5 kilofeet. The standard deviation for the projected intercept miss is plotted as $\sqrt{x_o^{(1)}}$. This curve shows that the probability density function of the projected intercept miss first expands because U is using greater positive control than V. However, around t = 3 seconds, V start to apply greater positive control while U control becomes negative, the probability density function contracts rapidly, the players are moving towards one another. The curve goes through a minimum and start rising again, this in effect tells us that V has overshot the inside maneuvering attacking missle after t = 7 seconds.

Figure 10 presents the maxmin effective lateral velocities for each player computed from the following equations:

$$\overline{c_u^2(t)} = u^2(t)\ x_2(t) \qquad \text{...........(4.55a)}$$

$$\overline{c_v^2(t)} = v^2(t)\ x_o(t) \qquad \text{...........(4.55b)}$$

These curves show that both quantities are well within the lateral velocities limit of 1.5 kilofeet/second for U and .75 kilofeet/second for V. The important point here is that U is much bolder for the maxmin game than V since in this situation, it is V who must play his "security level" control and guard against any possibility that U

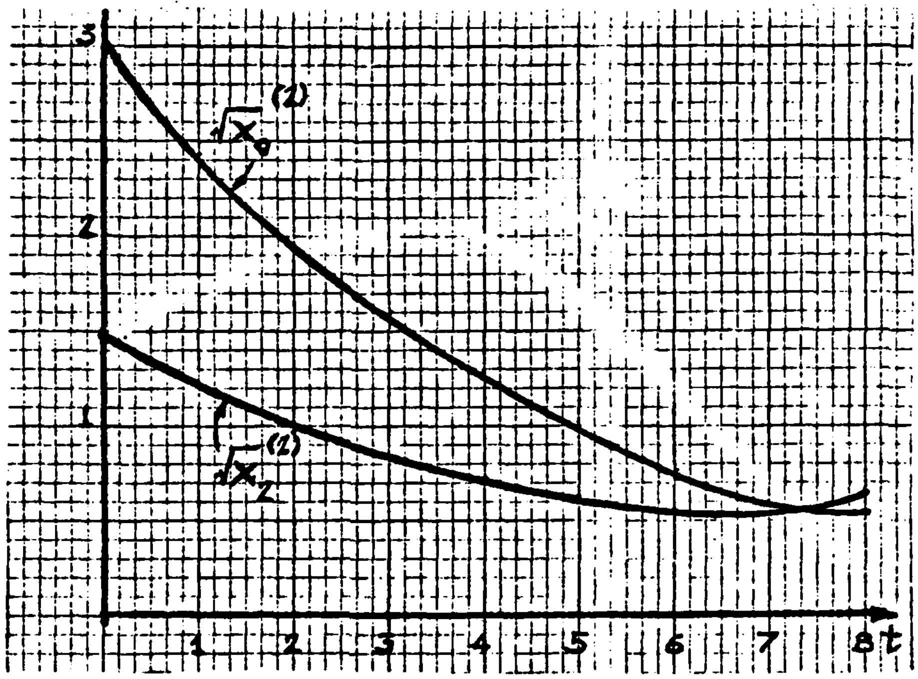


Figure 11. Maxmin Standard Deviation for Projected Intercept and Target Miss, Minimum No. 2

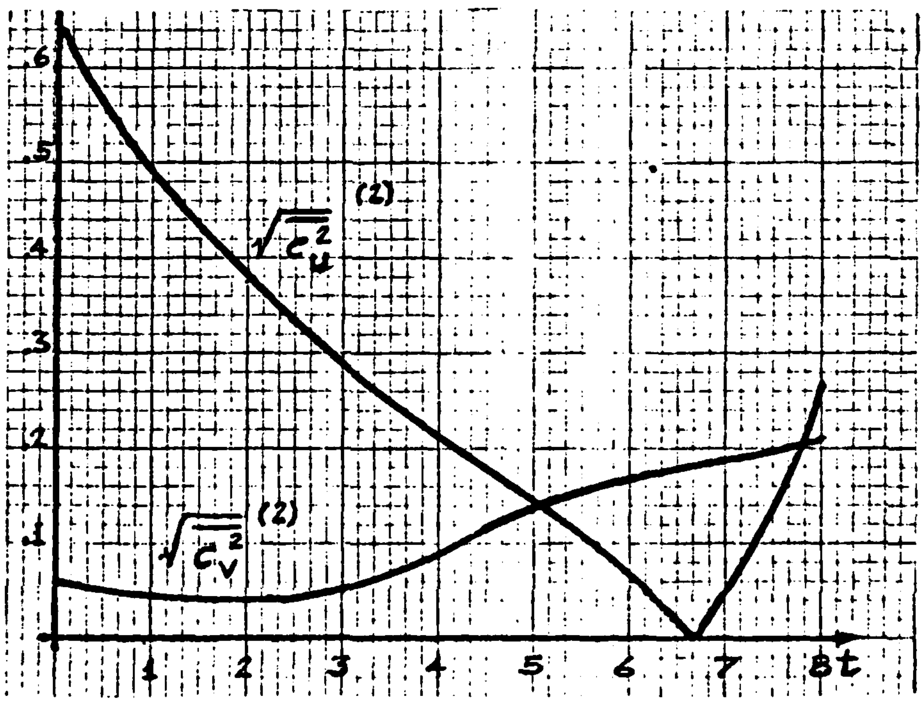


Figure 12. Maxmin Effective Lateral Velocities, Minimum No. 2

might come up with.

Figure 11 and 12 presents the same information as Figure 9 and 10 respectively, only this time with the control combination of $\tilde{v}(t)$ and $\tilde{u}^{(2)}(t)$. The negative feedback causes both standard deviations to drop initially until t = 6.6 seconds. After which time $\sqrt{x_2^{(2)}}$ increases becauses of positive value of $\tilde{u}^{(2)}(t)$ and $\sqrt{x_o^{(2)}}$ is level off a little because now U is trying to "get away" from V rather than just "bearing down" on the target. From Figure 12 V is again much more conservative on the lateral velocity than U.

Figure 13 shows sample trajectories with control combinations $\tilde{v}(t)$, $\tilde{u}^{(1)}(t)$ for Figure 13a, and $\tilde{v}(t)$,$\tilde{u}^{(2)}(t)$ for Figure 13b. These trajectories are generated by equations (4.8) using $\sigma_u(0)$ and $\sigma_v(0)$ as the initial values for the random variables x_u and x_v respectively. Clearly, the control combination $\tilde{v}(t)$, $\tilde{u}^{(1)}(t)$ represent the case where U initially maneuvering away from the target to draw V out, then using his superior control capability to maneuver inside. Whereas the control combination $\tilde{v}(t)$ and $\tilde{u}^{(2)}(t)$ represent the case where U first tries to bear down on the target and then uses his superior capability to maneuver outside to get away from V at the line of intercept.

4.4.3 Minmax Solution

As before, we need a nonimal $u_o(t)$ to start the

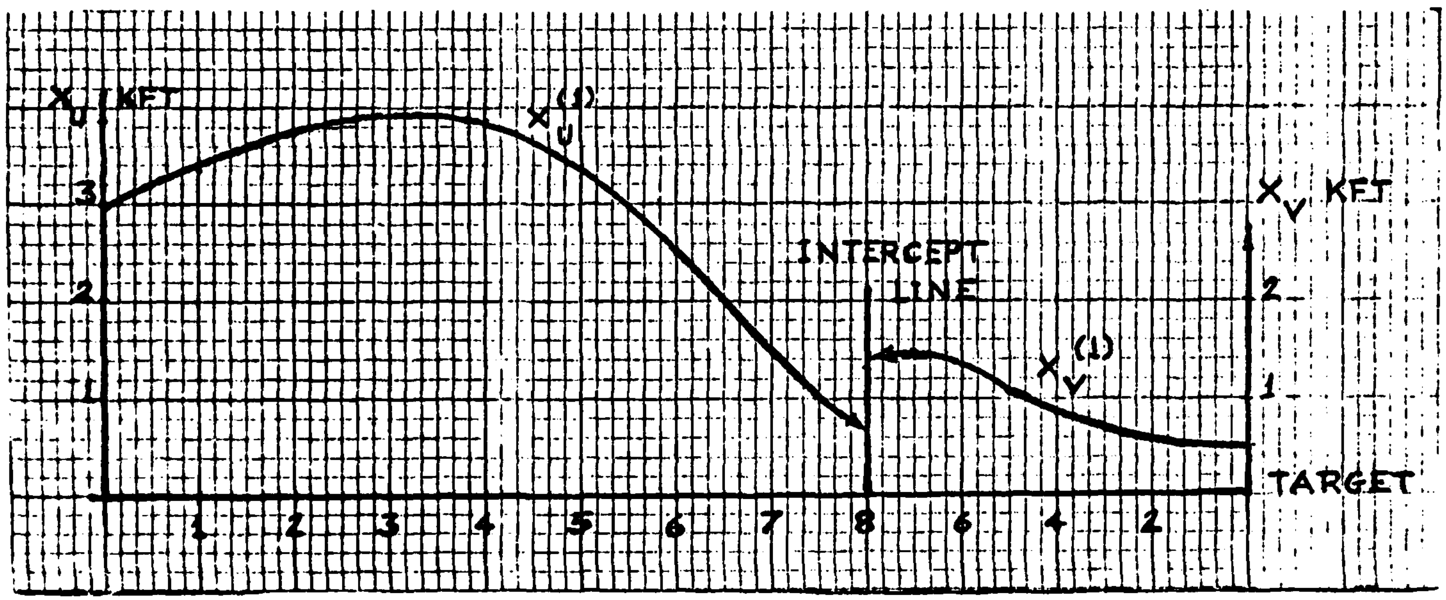


Figure 13a. Sample Trajectory of Maxmin Solution No. 1

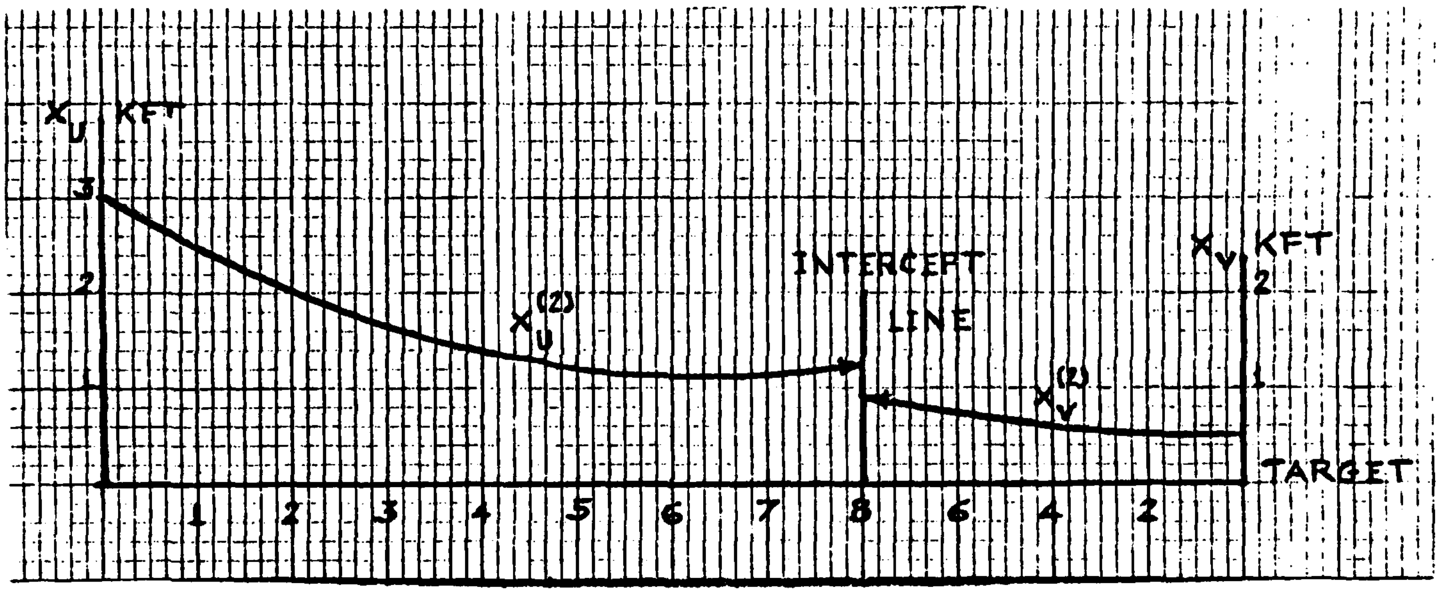


Figure 13b. Sample Trajectory of Maxmin Solution No. 2

algorithm for the minmax solution. Again we shall derive this nominal control by assuming $v_o(t) = 0$ for $0 \leqslant t \leqslant T$. With U using this control $u_o(t)$, V would be forced to do "something" to try maximize the cost functional.

Proceeding in the same manner as in the maxmin case, it is found that with $v_o(t) = 0$, $u_o(t)$ = constant. With this fact in mind the state and costate equations can be solved as before and substitue in

$$u_o = [2x_1(t)\lambda_o(t)+kx_1(t)\lambda_1(t)+x_2(t)\lambda_1(t)+2kx_2(t)\lambda_2(t)]/2Q_1 \qquad (4.56)$$

The resulting transcendental equation, even though more complicated, possesses the same structure as equation (4.47). Numerically solving this transcendental equation, u_o=-0.125 is used as the starting nominal control for the minmax algorithm.

The next step is to find all the local maxima of $J(u_o,v)$ using the DDP routine whose pertinent equations are as follows:

$$v^*(t) = (2Wv_o-2x_oJ_{x_o} - x_1J_{x_1})/2(Q_2+W) \quad \dots\dots\dots\dots\dots(4.57)$$

$$\dot{a}(t) = Q_2(v^*- v_o)^2 \quad \dots\dots\dots\dots\dots(4.58a)$$

$$\dot{J}_{x_o}(t) = 2v^*J_{x_o} \quad \dots\dots\dots\dots\dots(4.58b)$$

$$\dot{J}_{x_1}(t) = -2u_oJ_{x_o}-(ku_o-v^*)J_{x_1} \quad \dots\dots\dots\dots\dots(4.58c)$$

$$\dot{J}_{x_2}(t) = -u_o(J_{x_1}+2kJ_{x_2}) \quad \dots\dots\dots\dots\dots(4.58d)$$

Equation (4.57) is used for $|v^*| \leq 1$. If $|v^*| > 1$ then set $|v^*| = 1$ while preserving the same sign and use equation (4.52) instead of equation (4.58a). Incidentally, the terminal conditions on equations (4.58) are the same as equations (4.51). Successful backward integration of equations (4.58) while solving for v^* in (4.57) depends upon the proper value of the weighting factor W for the convergence control. For the problem at hand DDP routine converges to one maximum value from a wide selections of starting nominal control $v_o(t)$. Intuitively, once U's control is specified, then his trajectory is predictable from equation (4.8a). Knowing U's trajectory then V can aim at the position on the interception line and just minimizes the intercept miss which in turn maximizes the probability of survival of the target. It is reasonable to expect a unique optimal control for V to achieve this objective.

Table 12 summarizes the numerical results for the minmax iterations. For $u_o = -0.125$ and $v_o = 0.125$, the cost J = .758, the predicted cost change is .003 while the actual cost change is 0.145. The new cost J is .903 and v^* is accepted as v_o for the next iteration and so on. For the current u_o, DDP routine converges in seven iterations.

The maxcost gradient designated as g_o is then computed for the control combination u_o, v_o^*. The norm of this max-cost gradient is computed from

Table 12. Minmax Iteration

MIN	MAX	J	W	a(0)	ΔJ
u_o	v_o	.758	1.00	2.98E-3	0.145
		.903	0.90	3.30E-3	5.00E-2
		.953	0.81	4.40E-4	3.00E-2
		.956	0.73	1.50E-4	1.00E-4
		.956	0.66	3.10E-4	2.90E-3
		.959	0.59	7.00E-5	8.00E-5
		.959	0.53	1.80E-4	1.00E-3
	v_o^*	.960			
$(g_o,g_o) = 0.0119$, STEP $= 1.00$					
u_1	v_1	.822	1.00	2.67E-3	5.40E-2
		.936	0.90	2.00E-A	2.00E-3
		.938	1.46	7.00E-5	1.00E-3
	v_1^*	.939			
$(g_1,g_1) = 0.1019$, STEP $= 1.00$					
u_2	v_2	.761	1.00	9.00E-5	1.60E-3
		.762	0.90	8.00E-5	1.95E-3
		.764	0.81	8.00E-5	1.00E-3
		.765	0.729	7.00E-5	1.00E-3
		.766	0.656	6.00E-5	1.00E-3
	v_2^*	.766			
$(g_2,g_2) = .0075$, MINMAX SOLUTION					

$$(g_o, g_o) = \int_0^T g_o^2(t)\,dt \qquad \text{..........(4.59)}$$

The step length is computed using the same logic as in the maxmin case. The new nominal control combination is

$$u_1(t) = u_o(t) - \text{step}.g_o(t) \qquad \text{..........(4.60a)}$$

$$v_1(t) = v_o^*(t) \qquad \text{..........(4.60b)}$$

The process is then repeated. After two approximations for outer minimization, the norm of the maxcost gradient is negligibly small and the minmax solution is found with

$$\hat{u}(t) = u_2(t) \qquad \text{...........(4.61)}$$

The value of u(t) is shown in table 13 for sixty-four incremented time scale. The successively improved approximations to the minmax control are plotted in Figure 14a, while the corresponding maximizing controls are plotted in Figure 14b.

Figure 15 presents pertinent state information by showing standard deviation for projected intercept and target miss. The probability density function for the target miss initially drops sharply as shown by $\sqrt{x_o}$ caused by the fact that the players (using minmax control combination) are moving towards one another in this time interval. The positive control $\hat{u}(t)$ for $4.4 \leq t \leq 8$ not only slow down the rate of decrease of the intercept miss but also increase the target miss as U is moving away from the LOS.

Figure 16 shows the reversal of roles between U and V.

MINMAX SOLUTION FOUND COST = 0.7[illegible]

TIME	USTAR	VSTAR
1	-0.09266	0.20906
2	-0.09114	0.20890
3	-0.08955	0.20872
4	-0.08792	0.20854
5	-0.08621	0.20833
6	-0.08444	0.20811
7	-0.08260	0.20788
8	-0.08068	0.20763
9	-0.07870	0.20737
10	-0.07664	0.20709
11	-0.07451	0.20679
12	-0.07230	0.20648
13	-0.07001	0.20616
14	-0.06764	0.20581
15	-0.06519	0.20545
16	-0.06266	0.20508
17	-0.06004	0.20469
18	-0.05734	0.20429
19	-0.05454	0.20387
20	-0.05166	0.20344
21	-0.04869	0.20300
22	-0.04563	0.20255
23	-0.04248	0.20209
24	-0.03924	0.20162
25	-0.03590	0.20114
26	-0.03247	0.20066
27	-0.02895	0.20017
28	-0.02533	0.19969
29	-0.02162	0.19920
30	-0.01782	0.19873
31	-0.01392	0.19826
32	-0.00993	0.19780
33	-0.00585	0.19736
34	-0.00167	0.19693
35	0.00259	0.19653
36	0.00695	0.19616
37	0.01139	0.19581
38	0.01592	0.19551
39	0.02053	0.19524
40	0.02522	0.19503
41	0.03000	0.19487
42	0.03485	0.19476
43	0.03978	0.19473
44	0.04479	0.19476
45	0.04986	0.19488
46	0.05500	0.19509
47	0.06020	0.19539
48	0.06547	0.19580
49	0.07079	0.19632
50	0.07616	0.19697
51	0.08158	0.19775
52	0.08704	0.19867
53	0.09254	0.19976
54	0.09807	0.20102
55	0.10362	0.20246
56	0.10920	0.20411
57	0.11479	0.20597
58	0.12038	0.20807
59	0.12597	0.21043
60	0.13155	0.21307
61	0.13711	0.21601
62	0.14265	0.21929
63	0.14814	0.22293
64	0.15359	0.22697

Table 13. Computer Printout of Minmax Solution

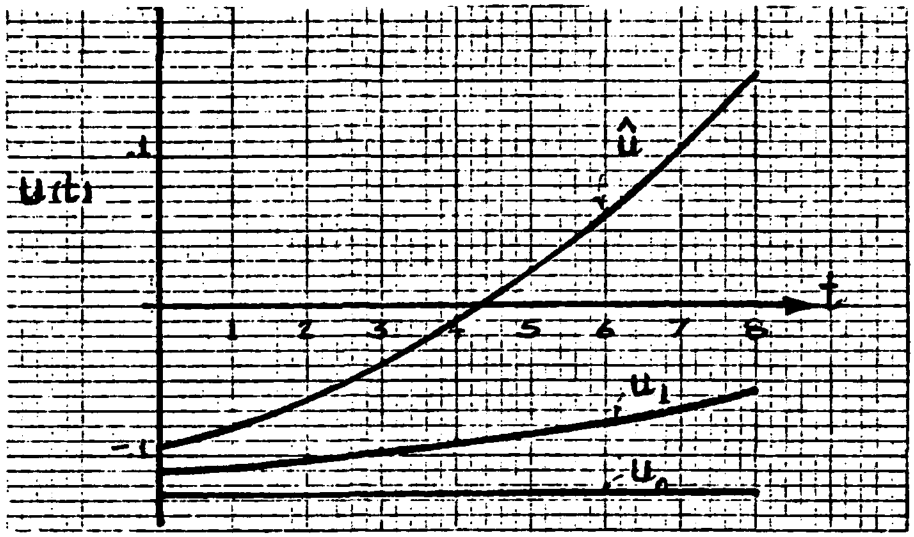


Figure 14a. Successive Approximations to Minmax Control

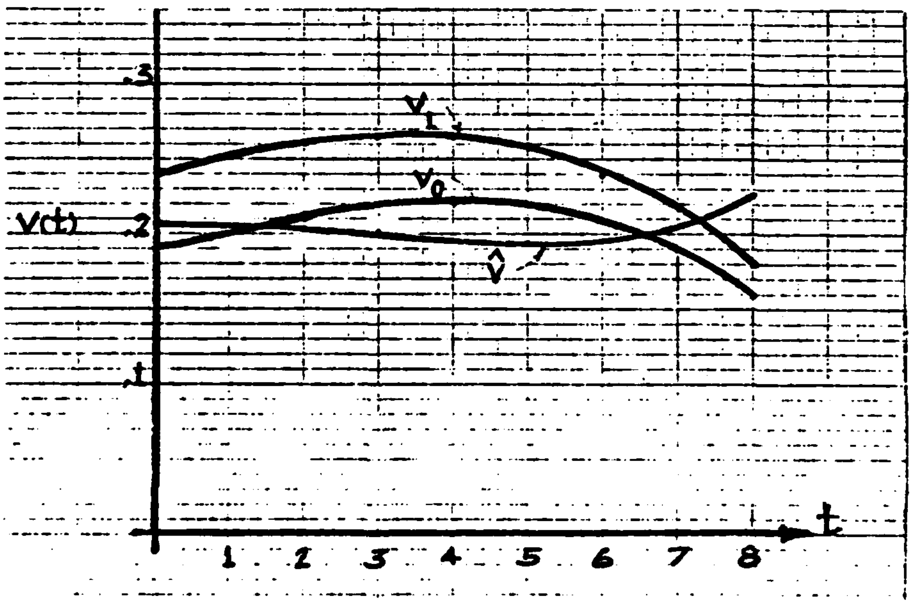


Figure 14b. Locally-Maximizing v(t) for each Successive Approximation

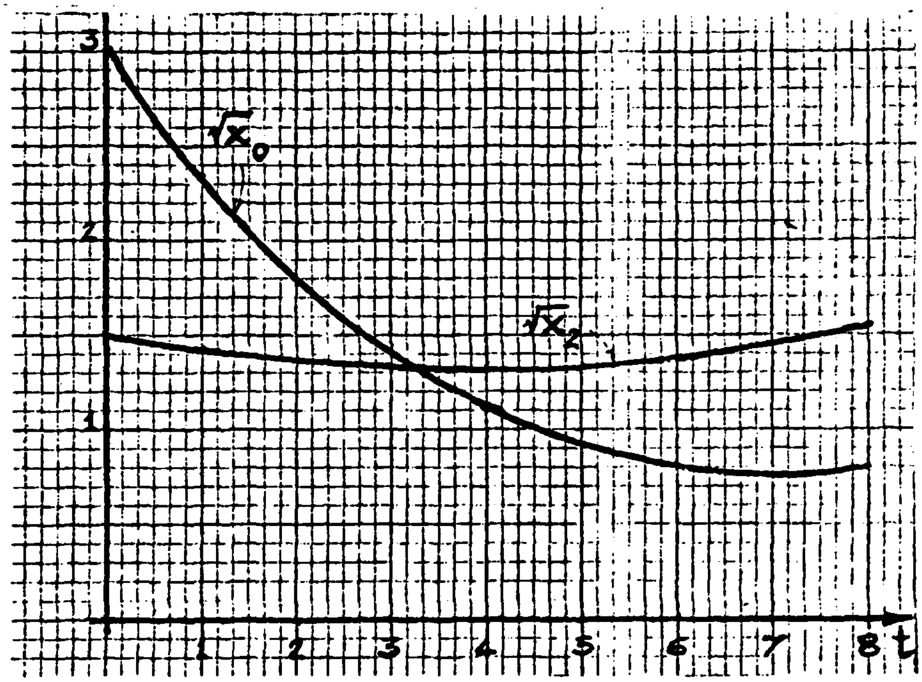


Figure 15. Minmax Standard Deviation for Projected Intercept and Target Miss

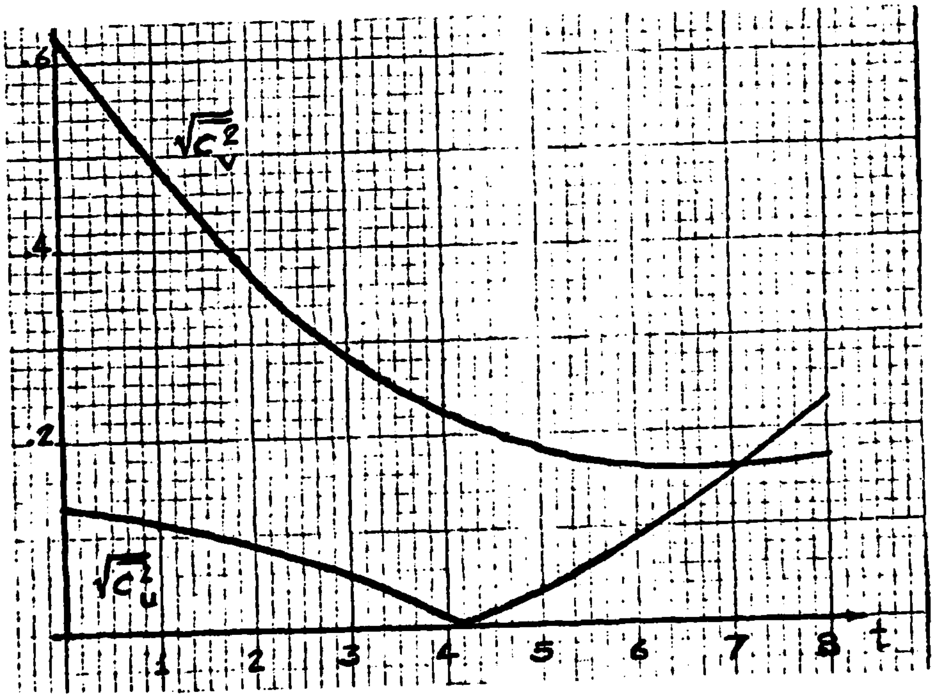


Figure 16. Minmax Effective Lateral Velocities

In the maxmin solution U was clearly the more agressive player, while in this case for the minmax solution U has to protect himself against all possibilities and becomes much more conservative than V.

This fact is also confirmed in Figure 17 which is a sample trajectory for minmax solution. U stayed pretty close to his initial lateral position while V is considerably bolder in going out to meet the attacking missle. For the minmax case, the players pass the intercept line with greater lateral distance away from the target than in the maxmin case.

4.4.4 Net Solution

The net solution of the game consists of the minmax control $\hat{u}(t)$ *for the attacking* missle player U and the maxmin control $\tilde{v}(t)$ for the intercepting missle player V. Using these strategies neither player needs to assume any prior knowledge on what the other player might do. The inequalities (2.7) also assures that both players will bene-benefit by using these "secure strategies".

The net cost of the game $J(\hat{u},\tilde{v})$ is 0.756, an improvement for U for the minmax cost of 0.766, and an even more improvement for V for the maxmin cost of 0.659.

Figure 18 shows a sample trajectory using the control combination $\hat{u}(t)$ and $\tilde{v}(t)$ generated by equations (4.8) using standard deviations $\sigma_u(0)$ and $\sigma_v(0)$ for the initial values $x_u(0)$ and $x_v(0)$ respectively.

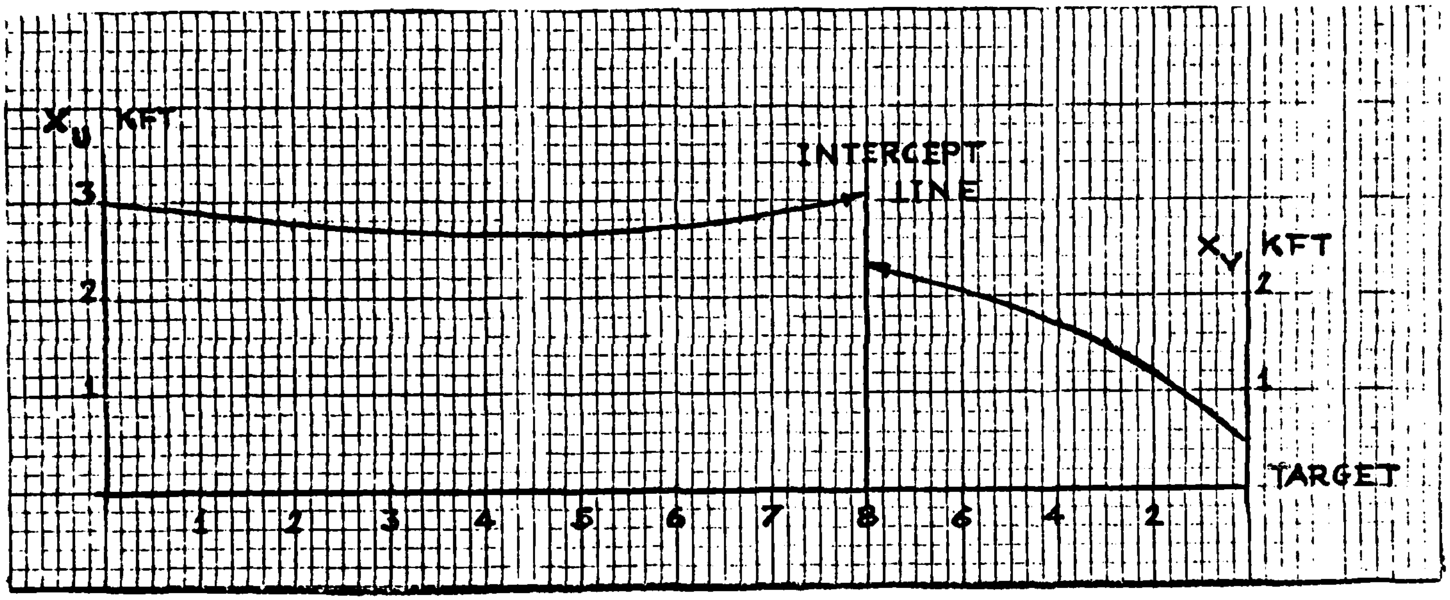


Figure 17. Sample Trajectory for Minmax Solution

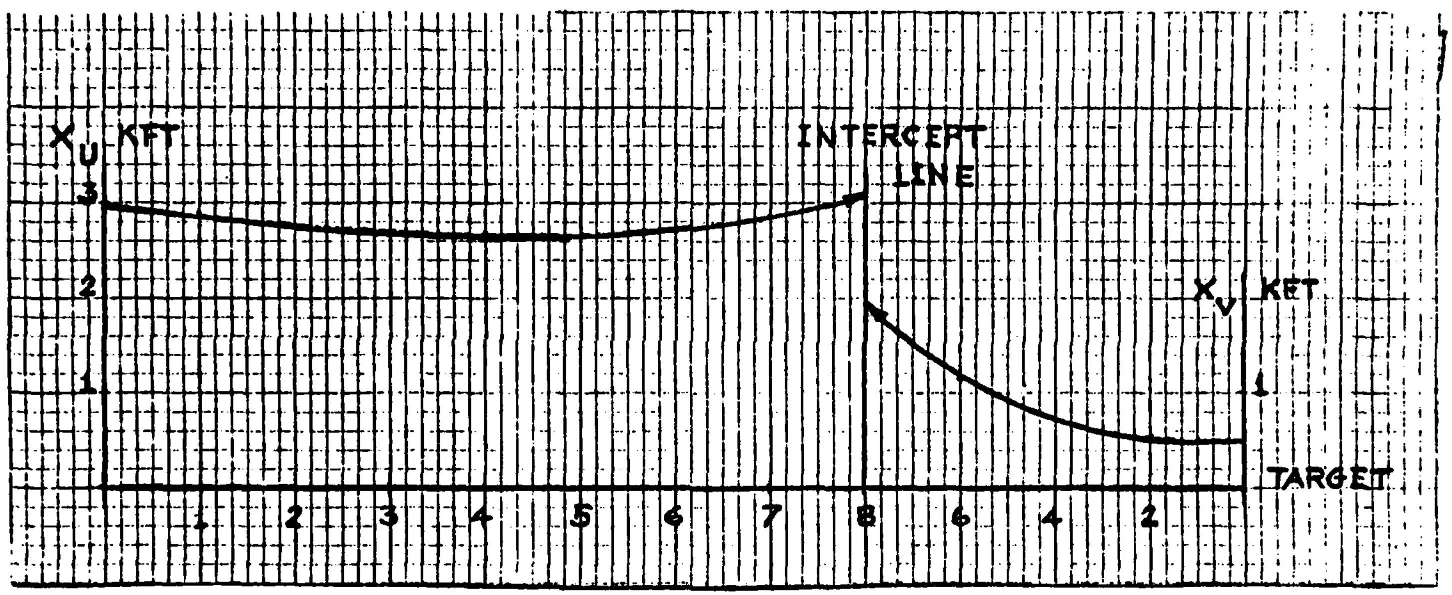


Figure 18. Smaple Trajectory for Net Solution

4.5 Discussion of the Problem

This problem has raised several significant points about general nature of differential games and the relative merit of the methods required to solve them. These points will be discussed in this section. First the results of this problem confirm that a saddle point is not required to exist in a general two-person zero-sum differential game. In this case, the maxmin solution is a crossover point. It must be emphasized, however, that a crossover point can be used as a solution only in the case where the values of the local optimal points in the inner optimization are moving towards one another after the improved successive approximations of the outer optimization process. Only a very limited class of differential game can be shown to possess a saddle point. One such problem is a linear dynamics, quadratic cost, two-person, zero sum deterministic differential game.

The seemingly simple problem described in this chapter with several simplified assumptions has turned out to be a complicated problem with nonlinear dynamics and nonquadratic cost. Even the initial guess of the initial nominal control solving as an optimal control problem cannot be treated analytically. The only way out was to use an efficient algorithm to solve the problem numerically. The use of a computer is inevitable

The defensive aspect of using the minmax and the maxmin solutions or the so called "security level" solutions must be mentioned. With these strategies, each player can rest assure that his opponent can induce no more harm to him than what he expects. He can only gain if the opponent decides to switch to some other strategy. One traditional way to solve an intercept problem was to derive a likely strategy for the attacking missle. Many authors have used colored-noise process to describe the behavior of the attacking missle and solved for the interception strategy as an optimal control problem. It is unreasonable, however, to expect that the attacking missle will oblige in the actual case and behave like a colored-noise process especially if he knows that the interception strategy has been derived in such manner. Some authors suggested "mixed strategy" as the solution of a differential game where a saddle-point does not exist. However, the "mixed strategy" solution is very hard to compute even for a very very simple unrealistic problem. Implementation of such strategy in an actual combat encounter does seem to be too far fetched.

Sufficient statistics are used as states variables for this problem to obtain meaningful results. This method greatly reduced the complexity of stochastic differential game. However, the number of state variables are greater than the deterministic cases. The current pursuit-evasion problem requires two physical state variables, the actual sufficient statistics is five: two mean value functions and three independent covariance matrix elements. If the physical controls are accelerations rather than velocities, four physical states variables would be required leading to fourteen elements in the set of sufficient statistics, and so on. To keep the problem tractable and retain physical insight, the problem should be simplified as much as realistically possible.

One reason that causes differential games to be much more complicated than simply being an extension of optimal control problems is the existence of conjugate points. The main reason conjugate points appear in differential games as a rule rather than an exception is that one must simultaneously maximizing and minimizing the same cost function. The first order algorithm developed for this report at first does not converge on a wide varieties of starting initial conditions even with the use of the "step -size" convergence control discussed in Chapter 2 which was used successfully in McFarland algorithm. Jarmark's scheme for convergence control is then used with excellent results. Singular control becomes non-singular with the proper convergence control weight described in section 4.3.

The formulation of the stochastic nonlinear pursuit-evasion problem in this chapter turns out to be the same model that McFarland used in his report. However, different rationalization was made in the formulation process. It is believed that the rationalization used here is more realistic as explained in section 4.2. Since the same model was arrived at except for the addition of the control constraints here, comparison of the computation results could be made. Since McFarland used second order algorithm for inner optimization and also computed a new estimate for the inner optimal control for each successive outer approximation, the amount of computations required for each iteration of McFarland's DDP is about four times the amount of computations required for each iteration used in this report. On this basis, the amount of computations required to reach the maxmin solution in this report is about one-half the amount required by McFarland's algorithm. The minmax computation requirement is even more impressive, only 6.35 seconds is required for the computer execution time while less than one-sixth the amount of computation is required when compared to McFarland's method to reach the same solution. Therefore, the algorithm presented in this report is more suitable to the real time application of pursuit and evasion differential game.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This research deals mainly with the most natural applications of differential games: namely the pursuit-evasion problems. Except for a very few simple unrealistic problems of this type, analytical solutions are virtually impossible to obtain. Fast and efficient algorithm is needed before the solutions of realistic pursuit-evasion problems can be solved and implemented in the actual physical conditions.

A numerical first order method without a saddle point assumption and capable of handling the control and state constraints is developed in this report. The algorithm is used to solve for the minmax and the maxmin solutions independently to be used by each player.

Linear Quadratic differential game without any constraint can be solved analytically even without a saddle point assumption. The analytical solutions are offered in this case. The case with limiting control constraint cannot be solved analytically. The numerical methods with and without the saddle point assumption give the answer for this case with negligible difference in computation time.

A nonlinear stochastic pursuit-evasion problem is developed and treated as a deterministic problem via a set of sufficient statistics. This problem does not converge with the algorithm developed in this report without an efficient convergence control method. The step-size convergence control is not adequate for this problem. Jarmark's convergence control method solves this problem when used in conjunction with the first-order algorithm developed here. Together, the algorithm provide a fast computation time for the problem. The results of this problem also strenghten the claim that saddle point does not exist in a general differential game contrary to the assumption used by many authors.

On the computation details, we found that the simple Euler integration scheme is sufficient when used in a simple linear-quadratic problem presented in Chapter 3. In the nonlinear stochastic case of Chapter 4, however, more accurate integration scheme is needed. The fourth-order Runge-Kutta integration scheme is used in this latter case with excellent results.

5.2 Recommendations for Future Research

The first fasinating area that could be further explored in this field is the improvement of convergence control method. Jarmark's convergence control technique may be applied to the second order algorithm developed by McFarland and compared with the "step-size" convergence control of Jacobson and Mayne. So far Jarmark has provided only the existence Theorems of the convergence control weighting matrix W. There is no hard and fast rule on how to find the most efficient W. This author feels that the convergence index domain used in this report has been started off in the right direction. There could be room for improvement in this area.

Another unexplored area is to let W be time variant. This would probably be suitable for the algorithm developed in this report. In the backward integration of the DDP equations, it does seem that the optimal control generated near the final time T should be more accurate than that generated further out near the starting time 0. Therefore, we could for example let W be a linear function of time with appropriate slope and initial value to be found by some suitable logic.

Time delay is another intriging area that must be faced in the real world. Presently, a few authors has dealt with this subject. All of them quickly specialize into simple problem. It would be interesting and beneficial to see how the information time delay would effect the optimal strategies of the problem presented in this report.

END